DMR 2025

The 6th International Workshop on Designing Meaning Representations

Proceedings of the Workshop

August 4, 2025
Prague, Czechia

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker Street S.
Suite 400-134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-296-1

Preface

While deep learning methods have led to many breakthroughs in practical natural language applications, there is still a sense among many NLP researchers that we have a long way to go before we can develop systems that can actually "understand" human language and explain the decisions they make. Indeed, "understanding" natural language entails many different human-like capabilities, and they include but are not limited to the ability to track entities in a text, understand the relations between these entities, track events and their participants described in a text, understand how events unfold in time, and distinguish events that have actually happened from events that are planned or intended, are uncertain, or did not happen at all. We believe a critical step in achieving natural language understanding is to design meaning representations for text that have the necessary meaning "ingredients" that help us achieve these capabilities. Such meaning representations can also potentially be used to evaluate the compositional generalization capacity of deep learning models.

This workshop intends to bring together researchers who are producers and consumers of meaning representations and, through their interaction, gain a deeper understanding of the key elements of meaning representations that are the most valuable to the NLP community. The workshop will provide an opportunity for meaning representation researchers to present new frameworks and to critically examine existing frameworks with the goal of using their findings to inform the design of next-generation meaning representations. One particular goal is to understand the relationship between distributed meaning representations trained on large data sets using network models and the symbolic meaning representations that are carefully designed and annotated by NLP researchers, with an aim of gaining a deeper understanding of areas where each type of meaning representation is the most effective.

These proceedings include papers presented at the 6th International Workshop on Designing Meaning Representations on August 4, 2025 in Prague, Czechia. DMR 2025 received 9 submissions, out of which 6 papers have been accepted to be presented at the workshop as talks. The papers address topics ranging from meaning representation methodologies to issues in meaning representation parsing, to the adaptation of meaning representations to specific applications and domains, to cross-linguistic issues in meaning representation. In addition to oral paper presentations, DMR 2025 also featured invited talks by Roberto Navigli (Sapienza University of Rome) and Mehrnoosh Sadrzadeh (University College London), entitled "NounAtlas, VerbAtlas, BMR, MOSAICo and other marvels: Towards a Unified Multilingual Semantic Framework" and "Quantum machine learning for natural language processing," respectively.

We thank our organizing committee for its continuing organization of the DMR workshops. We are grateful to all of the authors for submitting their papers to the workshop and our program committee members for their dedication and their thoughtful reviews. Finally, we thank our invited speakers for making the workshop a uniquely valuable discussion of linguistic annotation research.

Kenneth Lai and Shira Wein

Workshop Chairs

Kenneth Lai, Brandeis University
Shira Wein, Amherst College

Organizing Committee

Jan Hajič, Charles University
Hana Kubištová, Charles University
Alexis Palmer, University of Colorado Boulder
Martha Palmer, University of Colorado Boulder
James Pustejovsky, Brandeis University
Zdeňka Urešová, Charles University
Nianwen Xue, Brandeis University

Program Committee

Omri Abend, Hebrew University of Jerusalem
Maxime Amblard, Université de Lorraine, CNRS, Inria, LORIA
Zahra Azin, Carleton University
Ayoub Bagheri, Utrecht University
Katrien Beuls, Université de Namur
Abhidip Bhattacharyya, University of Massachusetts Amherst
Claire Bonial, Army Research Lab
Johan Bos, University of Groningen
Richard Brutti, Brandeis University
Maja Buljan, University of Oslo
Alastair Butler, Hirosaki University
Jayeol Chun, Brandeis University
Valeria de Paiva, Topos Institute
Lucia Donatelli, Vrije Universiteit Amsterdam
Katrin Erk, University of Texas at Austin
Kilian Evang, Heinrich Heine University
Federico Fancellu, Samsung Artificial Intelligence Center
Frank Ferraro, University of Maryland, Baltimore County
Annemarie Friedrich, University of Augsburg
Bruno Guillaume, Université de Lorraine, CNRS, Inria, LORIA
Jan Hajič, Charles University
Daniel Hershcovich, University of Copenhagen
Jena Hwang, Allen Institute for AI
Elisabetta Jezek, University of Pavia
Paul Landes, University of Illinois Chicago
Alex Lascarides, University of Edinburgh
Bin Li, Nanjing Normal University
Markéta Lopatková, Charles University
Marie McGregor, University of Colorado Boulder
Adam Meyers, New York University
Sarah Moeller, University of Florida
Massimo Moneglia, University of Florence
Skatje Myers, University of Wisconsin Madison
Juri Opitz, University of Zurich
Martha Palmer, University of Colorado Boulder
Weiguang Qu, Nanjing Normal University

Nathan Schneider, Georgetown University
Djamé Seddah, Inria Paris
Kevin Stowe, Educational Testing Service
Haibo Sun, Brandeis University
Zdeňka Urešová, Charles University
Rossella Varvara, University of Turin
Clare Voss, Army Research Lab
Susan Windisch Brown, University of Colorado Boulder
Nianwen Xue, Brandeis University
Annie Zaenen, Stanford University
Deniz Zeyrek, Middle East Technical University
Leixin Zhang, University of Twente
Xiao Zhang, University of Groningen
Jin Zhao, Brandeis University

Invited Speakers

Roberto Navigli, Sapienza University of Rome
Mehrnoosh Sadrzadeh, University College London

Table of Contents

Workshop Program

Monday, August 4, 2025

8:50–9:20 ***Check-in and Coffee Hour***

9:20–9:40 ***Opening Remarks***

9:40–10:40 ***Invited Talk by Roberto Navigli: NounAtlas, VerbAtlas, BMR, MOSAICo and other marvels: Towards a Unified Multilingual Semantic Framework***

10:40–11:10 ***Coffee Break***

11:10–11:30 *Comparing Manual and Automatic UMRs for Czech and Latin*
Jan Štěpánek, Daniel Zeman, Markéta Lopatková, Federica Gamba and Hana Hledíková

11:30–11:50 *The Role of PropBank Sense IDs in AMR-to-text Generation and Text-to-AMR Parsing*
Thu Hoang, Mina Yang and Shira Wein

11:50–12:10 *Boosting a Semantic Parser Using Treebank Trees Automatically Annotated with Unscoped Logical Forms*
Miles Frank and Lenhart Schubert

12:10–12:30 *Using MRS for Semantic Representation in Task-Oriented Dialogue*
Denson George, Baber Khalid and Matthew Stone

12:30–13:30 ***Lunch***

13:30–14:30 ***Discussion: The Role of Semantic Representations in the Age of Large Language Models***

14:30–15:00 ***Coffee Break***

15:00–15:20 *Evaluation Framework for Layered Meaning Representation*
Rémi de Vergnette, Maxime Amblard and Bruno Guillaume

Monday, August 4, 2025 (continued)

15:20–15:40 *Representing ISO-Annotated Dynamic Information in UMR*
Kiyong Lee, Harry Bunt, James Pustejovsky, Alex C. Fang and Chongwon Park

15:40–16:40 ***Invited Talk by Mehrnoosh Sadrzadeh: Quantum machine learning for natural language processing***

16:40–17:00 ***Closing Remarks***

18:30–21:00 ***Dinner***

Comparing Manual and Automatic UMRs for Czech and Latin

Jan Štěpánek and Daniel Zeman and Markéta Lopatková and Federica Gamba and Hana Hledíková
Charles University, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics
Malostranské nám. 2/25, 118 00 Prague 1, Czechia
{stepanek,zeman,lopatkova,gamba,hana.hledikova}@ufal.mff.cuni.cz

Abstract

Uniform Meaning Representation (UMR) is a semantic framework designed to represent the meaning of texts in a structured and interpretable manner. In this paper, we evaluate the results of the automatic conversion of existing resources to UMR, focusing on Czech (PDT-C treebank) and Latin (LDT treebank). We present both quantitative and qualitative evaluations based on a comparison between manually and automatically generated UMR structures for a sample of Czech and Latin sentences. The findings indicate comparable results of the automatic conversion for both languages. The key challenges prove to be the higher level of semantic abstraction required by UMR and the fact that UMR allows for capturing semantic structure in multiple ways, potentially with varying levels of granularity.

1 Introduction

The challenge of representing meaning has been fascinating linguists, philosophers, and cognitive scientists for centuries. Traditional semantic frameworks—such as truth-conditional semantics (e.g., Davidson, 1967), frame semantics (e.g., Baker et al., 1998; Fillmore et al., 2002), and cognitive semantics (e.g., Langacker, 1987; Croft and Cruse, 2004)—aimed to formalize how meaning is constructed, interpreted, and communicated.

Recent advances in natural language processing have been driven by large language models. These models excel at downstream tasks such as text generation and translation. However, given their unclear interpretability—as they rely on statistical patterns rather than true semantic or logical understanding—they do not answer the essential questions about meaning representation.

Thus, symbolic approaches remain central to efforts to search for precise, inference-capable meaning representations. *Uniform Meaning Representation* (UMR), the fundamentals described by van Gysel et al. (2021), is one of the responses to this interest. We build on this initiative and test the approach for representing Czech and Latin—inflected languages with rich morphology and free word order representing information-structural features (such as topic-focus articulation and discourse dynamics) rather than syntactic relations. The results of our effort could thus provide valuable insight for the UMR community.

Creating data from scratch is extremely time-consuming and requires highly trained annotators with extensive expertise. That's why we aim to take advantage of the richly annotated datasets already available for the two languages, and investigate the possibility of their (semi-)automatic conversion to the UMR framework. Namely, we rely on the PDT-C corpus[1] (Hajič et al., 2024a) for Czech and on a subset of the Latin Dependency Treebank (LDT)[2] (Bamman and Crane, 2006) for Latin. Both are annotated using the same PDT annotation scenario, thus supporting the same conversion process. A similar approach has proved to be advantageous for English—as described by Bonn et al. (2023b), who created the extensive English UMR corpus (Bonn et al., 2025) from structures used in *Abstract Meaning Representation*, the UMR predecessor. Full conversion is not always feasible, but even partial results are highly beneficial, as shown by Buchholz et al. (2024) and Gamba et al. (2025).

The paper presents a comparison of (a small sample of) double-annotated UMR data, that is, the data with manually created UMR structures and their counterparts automatically converted from the PDT-C and LDT corpora, respectively. First, we briefly describe the UMR and PDT-C approaches and the available automatic conversion (§ 1.1, 1.2, and 1.3, respectively) and the Czech

[1] http://hdl.handle.net/11234/1-5813
[2] https://itreebank.marginalia.it/

Proceedings of the 6th International Workshop on Designing Meaning Representations, pages 1–12
August 4, 2025.

and Latin UMR data (§ 2). § 3 introduces the way we compare the structures and brings a quantitative comparison. A qualitative analysis follows in § 4. § 5 then summarizes the results and discusses further work.

1.1 Uniform Meaning Representation

Uniform Meaning Representation (see esp. van Gysel et al., 2021; Bonn et al., 2023b, 2024) is a semantic framework designed to represent the meaning of texts in an interpretable way, elaborating the (originally English-centered) *Abstract Meaning Representation* (Banarescu et al., 2013; Wein and Bonn, 2023). UMR's graph-based sentence-level representation abstracts from the overt sentence syntax; in particular, it encodes the frame-based predicate-argument structure of all eventive concepts, including their aspectual information. In addition, UMR models semantic relations that cross sentence boundaries, such as coreference, temporal chains, and epistemic modality, which makes it possible to interpret context and discourse more effectively.[3] Its applicability has been demonstrated on a sample of data from English, Chinese, and four low-resource American languages (Bonn et al., 2023a).

1.2 PDT: Deep syntactic representation

Both treebanks that we use as our source data, Czech PDT-C and Latin LDT, provide representation at the so-called deep syntactic layer (also tectogrammatical or t-layer; see esp. Sgall et al., 1986; Hajič et al., 2020 for Czech and Passarotti, 2014; Gonzalez Saavedra and Passarotti, 2014 for Latin). The core of this dependency-oriented representation is formed by the predicate-argument structure (valency) and other deep syntactic relations. This core structure is enriched with meaning-relevant morphological information (number and gender for nouns; tense, aspect, modality for verbs), topic-focus articulation, and coreference annotation.[4]

In contrast to UMR, the PDT scenario concentrates on linguistically structured meaning; as such, it more or less directly refers to the annotated text. Thus, this scenario is less abstract than UMR—which presents the main obstacles to the automatic conversion (as will be discussed below).

A more thorough comparison of the two approaches, envisaging the possibility of the automatic PDT-C to UMR conversion, can be found in Lopatková et al. (2024).

1.3 PDT to UMR automatic conversion

Here we work with the first attempt to automatically convert PDT structures to UMR structures, as described in Lopatková et al. (2025). Let us stress that this conversion is partial—it covers only selected phenomena pertaining to the sentence-level annotation (esp. structure of the graph, labeling of nodes and relations, PropBank-like argument structure for verbs, and selected attributes); in addition, intra-sentential coreference relations are identified.

The conversion procedure recursively traverses the PDT-C tree (namely the t-structure), and incrementally builds the corresponding UMR graph. Each node and edge are analyzed to determine necessary structural and labeling changes, as well as the addition of UMR attributes.

- In this stage, *structural transformations* are a key part of the process. These typically arise from handling coreference (merging pronouns with their referents, reentrancies, inverse roles), coordination (esp. representing conjuncts and their shared dependents in a UMR-adequate structure), relative clauses (merging referential nodes and linking them semantically), and control or raising verbs (merging arguments across predicates), as sketched by Lopatková et al. (2024, 2025).
- Changes in *nodes labeling* reflect the shift from deep syntactic elements of PDT-C (identified as t-lemmata) to UMR concepts (entities, states, and processes).
- For *edges labeling*, deep syntactic roles of PDT-C are converted to UMR semantic relations, using (i) verb-specific mapping of arguments (whenever available, Hajič et al., 2024b) and (ii) default mapping of arguments and adjuncts (Lopatková et al., 2025).
- UMR nodes are enriched with selected UMR attributes, namely aspect, degree, polarity, quant, refer-number, and refer-person (Lopatková et al., 2025).
- *Nodes alignment* is gained from PDT-C.

In the following, we concentrate on evaluating the quality of conversion for the aforementioned phenomena. We exclude UMR attributes not listed above (i.e., wiki, modal-strength, mode, polite, and

[3] The UMR 0.9 specification as available here: https://github.com/umr4nlp/umr-guidelines/blob/master/guidelines.md

[4] For the full PDT-C documentation, see https://ufal.mff.cuni.cz/pdt-c/documentation.

	corpus	sentences	tokens	PDT / LDT nodes	UMR nodes (manual)	UMR nodes (automatic)
Czech	PDT	25	467	378	375	349
	PDTSC	50	374	321	442	305
	PCEDT	16	474	400	307	327
	total	91	1315	1099	1124	981
Latin	LDT	50	889	928	773	865

Table 1: Statistics for both manually and automatically annotated data.

quote), as well as all phenomena represented in the document-level annotation.[5]

2 Double-annotated UMR Data

2.1 Czech UMR data

The PDT-C corpus offers a large volume of Czech data spanning various genres. We selected a sample of six files from its development data for manual annotation. This sample covers key genres presented in PDT-C (written texts in both general journalistic and technical styles, as well as spoken data). Another selection criterion was that the files include specific linguistic phenomena where we anticipate problems during the conversion (e.g., not overtly expressed entities or events, selected types of special constructions, coordinated structures, complex coreferential chains, negation). Specifically, the selected texts are as follows:

- 25 sentences (2 documents) from the core PDT[6] subcorpus (Czech newspaper texts from 1992-94);
- 50 sentences from the PDTSC[7] subcorpus (spontaneous dialogs);
- 16 sentences (out of 37 sentences, 2 documents) from the Czech part of the PCEDT[8] subcorpus (Czech translations of the Penn Treebank-WSJ texts).

Table 1 provides more detailed statistics. It reveals that the WSJ texts from PCEDT are more complex (especially compared to spontaneous dialogs from PDTSC); thus, despite the lower number of PCEDT sentences, the sample data selected for manual annotation provide relatively balanced coverage of the genres represented in the corpus.

A small portion of the data (21 sentences with 255 tokens from PDT and PDTSC) were annotated by two human annotators in parallel; these data were used to estimate inter-annotator agreement (Table 2).

2.2 Latin UMR data

The corpus utilized in this study corresponds to a portion of the LDT as provided by the *Index Thomisticus Treebank* project[9] (Passarotti, 2019). Compared to the original version, this subset was refined at the syntactic layer and annotated from scratch at the semantic-pragmatic layer. It includes the entire *De coniuratione Catilinae* 'Conspiracy of Catiline' by Sallust along with excerpts from the works of Caesar and Cicero. For this work, we focus specifically on Sallust and select the first 50 sentences of his work, corresponding to the first five (out of 61) chapters of the text. We select these sentences as they are already part of the UMR 2.0 release. Table 1 provides basic data statistics.

3 Comparison: Global Perspective

3.1 Metrics for graph comparison

Quantitative comparison of semantic graphs is a non-trivial task because two representations of the same sentence may differ in the number of nodes, and the node identifiers (variables) typically differ, too. It is thus not obvious which nodes should be taken as corresponding to each other. If we can find the optimal node mapping between the two graphs, the rest of the task is easy. Properties of the graph can be expressed as a set of triples (x, y, z), where x is a node (now identifiable in both graphs), y is a name of a relation or an attribute, and z is another node (child node of the relation) or the value of the attribute. Similarity of two graphs can be expressed as the F_1 score of the triples.

UMR is a successor to AMR, and for AMR, the *smatch* metric (Cai and Knight, 2013) has emerged as the de-facto standard. It defines as optimal the mapping that maximizes F_1 of the resulting triples;

[5] The Czech and Latin UMR data described and compared in the paper are available through the Lindat repository, see http://hdl.handle.net/11234/1-5951.

[6] https://ufal.mff.cuni.cz/pdt3.5

[7] https://ufal.mff.cuni.cz/pdtsc2.0

[8] https://ufal.mff.cuni.cz/pcedt2.0/

[9] https://itreebank.marginalia.it/

UMR node mapping:					
Anot1 nodes	Anot2 nodes	mapped	recall	precision	F_1
228	221	215	94%	97%	96%

Concept and relation comparison (only mapped nodes):*					
Anot1 triples	Anot2 triples	match	recall	precision	F_1
633	644	595	94%	92%	93%

Concept and relation comparison:**					
Anot1 triples	Anot2 triples	match	recall	precision	*ju:mœʧ* = F_1
663	659	595	90%	90%	90%

Table 2: Manually double-annotated UMRs: quantitative comparison for Czech (PDT+PDTSC). (* Unmapped nodes are ignored. ** Unmapped nodes all counted as incorrect.)

the *smatch* algorithm employs hill-climbing with restarts to find an approximate solution to the optimization problem.

An alternative node mapping algorithm, called *AnCast*, has been proposed specifically for UMR (Sun and Xue, 2024). It has been shown to be more efficient and more accurate than *smatch*. The authors also define a number of partial metrics, such as Concept F_1 and Labeled Relation F_1, which improve interpretability of the results.

One of the improvements of UMR over AMR is that UMR annotation includes alignment of nodes to surface tokens. *Smatch* does not have the notion of word alignment; *AnCast* can use it if available, but it can work without it, too. Nevertheless, *AnCast's* ability to exploit alignment is limited. The token–node alignments can be $M : N$, with a node potentially mapped to a discontinuous set of tokens, while *AnCast* can currently process only continuous alignments. *AnCast* also compares concepts of the nodes to be mapped, and it tries to assess concept similarity rather than identity, although in a restricted manner. To achieve similarity > 0, one concept lemma must be substring of the other. This would recognize similarity between e.g., *fry* and *stir-fry*, but not between Czech *volit* 'to vote' and nominalized *volba* 'election'.

Both *smatch* and *AnCast* will map as many nodes as possible. If one of the graphs has more nodes that the other, remaining nodes will stay unmapped. If the graphs have the same number of nodes, every node will be mapped to a node in the other graph, even if they are clearly unrelated. This may occasionally improve the score when a random attribute occurs in both nodes, but it blurs the interpretation of the score. More importantly, we also want to use the mapping to eye-ball disagreement between annotators, and maximal node mapping is not helpful for that purpose. Therefore, we employ a third mapping algorithm called *ju:mœʧ*, which primarily maps nodes aligned to the same word(s), and for nodes without word alignment (which are a minority in UMR graphs) requires concept identity. As with *smatch* and *AnCast*, we assess similarity of other node attributes if needed to get a symmetric one-to-one mapping. An example comparing *ju:mœʧ* and *smatch* mappings is given in Appendix A.

Note that all scores in the present paper evaluate only the sentence-level graphs in UMR. The document-level relations (modal and temporal annotation, coreference) could be evaluated as triples using the same node mapping, but the current evaluation scripts do not support it.

3.2 Quantitative comparison

Comparison of manually double-annotated Czech data. First, to gain insight into the problem, we quantitatively analyzed, using *ju:mœʧ* scores, a small sample of manually double-annotated Czech data (21 sentences with 255 tokens, annotated by two annotators in parallel). The scores cover all concept instance triples, all relations between nodes, and selected node attributes. To be able to use the same setting for the manually double-annotated data and for the comparison of the manually and automatically created structures, we skip attributes whose values cannot be obtained from the source data (wiki, modal-strength) and not-yet-converted source attributes (mode, polite, quote). The results are shown in Table 2.

UMR node mapping:						
corpus	MAN nodes	AUTO nodes	mapped	recall	precision	F_1
PDT	375	349	284	76%	81%	78%
PDTSC	442	305	235	53%	77%	63%
PCEDT	307	327	244	79%	75%	77%
total	1124	981	763	68%	78%	72%

Concept and relation comparison (only mapped nodes):*						
corpus	MAN triples	AUTO triples	match	recall	precision	F_1
PDT	844	819	502	59%	61%	60%
PDTSC	622	633	352	57%	56%	56%
PCEDT	714	588	342	48%	58%	53%
total	2180	2040	1196	55%	59%	57%

Concept and relation comparison:**							
corpus	MAN triples	AUTO triples	match	recall	precision	*ju:mæʧ* = F_1	*smatch*
PDT	1082	916	502	46%	55%	50%	49%
PDTSC	1318	770	352	27%	46%	34%	37%
PCEDT	916	757	342	37%	45%	41%	51%
total	3316	2443	1196	36%	49%	42%	45%

Table 3: Czech UMRs: quantitative comparison of manual and automatic structures.
(* Unmapped nodes are ignored. ** Unmapped nodes all counted as incorrect.
MAN stands for the manual annotation, AUTO for the automatic conversion.)

The table shows that *ju:mæʧ* was able to successfully map 96% of Czech nodes, with the overall F_1 over 90%. However, it is important to note that these figures were obtained after thorough discussions and reconciliation of problematic cases; as such, they represent an upper bound for what the automatic conversion procedure could achieve. While the inter-annotator agreement is reasonably high in this experiment (though the available data sample is very small), the results still indicate that we cannot expect perfect agreement, given the nature of the UMR framework.

Comparison of manual and automatic UMR structures. A basic quantitative analysis with *ju:mæʧ* scores is provided in Tables 3 and 4. The same setting is preserved (i.e., the scores cover all concept instance triples, all relations between nodes, and the same set of node attributes).

The tables reveal relatively low agreement: only 78% of Czech nodes and 77% of Latin nodes were successfully mapped by *ju:mæʧ*. For these correctly mapped nodes, around 60% of the triples match (57% for Czech and 62% Latin). When all nodes are scored, the overall F_1 drops to 42% for Czech and 51% for Latin. The results are broadly consistent across both languages. For Czech, the most elaborated PDT subcorpus displays consistently better conversion results (*ju:mæʧ* reaching 50%), while the PDTSC subcorpus has a low recall, as discussed in § 4.1.

For comparison, Tables 3 and 4 also provide figures obtained by the *smatch* metric. For both languages, these figures are higher than those of *ju:mæʧ* (increase of 3% for Czech and 10% for Latin). Note that *smatch* uses a different nodes mapping algorithm and that it does not allow for excluding selected attributes.

4 Comparison: Analysis of Differences

Despite rather low results reported above, visual comparison of the graphs for individual sentences often yields a fairly good match. The basic structure typically aligns, and differences are mostly local (concepts, relation types, or local structure).

In this section, we focus on the main sources of disagreement and attempt to determine whether they stem from shortcomings in the conversion process, differing interpretations of the annotation guidelines, or even annotation errors (which can potentially be reconciled). Another possible explanation for the observed results lies in the nature of the UMR framework itself, which—as repeatedly

UMR node mapping:					
MAN nodes	AUTO nodes	mapped	recall	precision	F_1
773	865	629	81%	73%	77%

Concept and relation comparison (only mapped nodes):*					
MAN triples	AUTO triples	match	recall	precision	F_1
1820	1923	1168	64%	61%	62%

Concept and relation comparison:**						
MAN triples	AUTO triples	match	recall	precision	*ju:mœʧ* = F_1	*smatch*
2174	2367	1168	54%	49%	51%	58%

Table 4: Latin UMRs: quantitative comparison of manual and automatic structures. (* Unmapped nodes are ignored. ** Unmapped nodes all counted as incorrect. MAN stands for the manual annotation, AUTO for the automatic conversion.)

noted in its specification—allows for multiple valid annotations of the same meaning (as the comparison of two manual structures illustrates).

The main differences between automatic and manual UMRs lie in the fact that UMR is more abstract than PDT and, at the same time, allows alternative annotations. In particular, abstract predicates (§ 4.1), event-entity distinction (§ 4.2), and abstract entities (§ 4.3) proved to be challenging.

4.1 Abstract predicates

To foster cross-linguistic comparability of meaning representations, UMR introduces several types of abstract predicates (also called abstract rolesets). Among these, rolesets for nonprototypical predication, so-called implicit rolesets, and predicates for reification need special attention during conversion.

Rolesets for nonprototypical predication. UMR predicates for nonprototypical predication capture possession, location, property and object predication, and identity relationships (e.g., have-91 or belong-91 for possession or have-mod-91 for property predication). In PDT, the corresponding semantic content is represented with the overt verb, typically *být* 'be' or *mít* 'have'.[10] The current version of the conversion keeps the lexical predicates *být* 'be' or *mít* 'have', which, of course, is not in compliance with the UMR specification.

As an exemplification, consider the (shortened) PDT example (1) and its manually and automatically created UMR structures (both simplified). The use of the first abstract predicate have-place-91 does not affect the overall structure at the upper level (the only differences being the node and the relation labels, :ARG0 and :place instead of :ARG1 and :ARG2, respectively). However, the UMR-compliant manual annotation substantially differs from the straightforward PDT annotation when it comes to the representation of the interpersonal relation; it employs the have-rel-role-92 predicate, which captures *sister* as a person (:ARG2) who has a 'sister' relation (:ARG4) to the speaker (:ARG1).

(1) … *je tam sestra…*
'… there is (my) sister there…'

```
MAN:
(b / have-place-91
  :ARG2 (t / place)  'there'
  :ARG1 (p / person
        :ARG2-of (h / have-rel-role-92
                 :ARG1 (p2 / person
                       :refer-number singular
                       :refer-person 1st)
                 :ARG4 (s / sestra)))) 'sister'

AUTO:
(b / být-011  'be'
    :place (t / tam)  'there'
    :ARG0 (s / sestra))  'sister'
```

Next steps: Typical candidates for nonprototypical predication should be identified: (i) Among the valency frames (framesets) of the verbs *být* 'be' and *mít* 'have', identify those corresponding to UMR predicates for nonprototypical predication, together with adequate argument role mapping. (ii) Determine other candidates for possessive predication (e.g., *vlastnit*, 'own, possess', *patřit (někomu)* 'belong to', possessive pronouns, etc.). (iii) Find relational nouns underlying object predication. However, identification of all candidates for abstract

[10] In these contexts, *být* 'be' or *mít* 'have' are considered predicates, i.e., lexical verbs rather than auxiliaries, in Czech linguistics, with valency frames (PDT analogy to framesets) characterizing each of their senses.

predicates remains a challenging task.

Reifications. Reification, a process of converting a role (= a relation) into a concept, is another important UMR feature. From the conversion perspective, it represents an additional source of disagreement. See, e.g., the manual annotation of example (2), where the :frequency relation is changed to the have-frequency-91 predicate in the manual annotation, while the relation is preserved in the automatic conversion. Formally, the upper structure of the graph is the same, the only changes deal with nodes labeling (the lexical predicate *být-011* 'be' to the reification have-frequency-91) and relations labeling (the role :frequency to :ARG2).

(2) *… teď je to každý rok.*
'… now it's every year.'

```
MAN:
(f / have-frequency-91
   :temporal (t / teď  'now'
   :ARG1 (e / event)
   :ARG2 (r2 / rate-entity-91
           :ARG3 (t / temporal-quantity
                   :quant 1
                   :unit (r / rok)))) 'year'

AUTO:
(b / být-011  'be'
    :temporal (t / teď)  'now'
    :ARG1 (t2 / ten)  'it (refers to event e)'
    :frequency (r / rok  'year'
                 :mod (k / každý)))  'every'
```

Next steps: Again, while the identification of individual valency frames of *být* 'be', which often underlies such structures, appears to be challenging but doable, automatic recognition of other candidates for reification seems a too ambitious task. As the UMR specification suggests applying reification only if needed, this step can be postponed.[11]

Implicit rolesets. UMR is characterized by a list of implicit rolesets that specify various types of information, the most relevant being the following:

- They can identify meta-language information (e.g., publication-91, hyperlink-91, and street-address-91).
- The second group is formed by predicates that express quantitative observations (e.g., include-91 to represent subsets, as in *some of them*, *23% of voters*; range-91 for *more than 2 months*).
- Yet other implicit rolesets indicate special constructions, as, e.g., comparison (like resemble-91 for *be like John*).
- They can also identify dialog-related structures (e.g., request-confirmation-91 for *Okay?*; say-91 for identifying communication structure (who says what to whom)).

In general, the comparison has revealed that it is very difficult to automatically identify language material in PDT that corresponds to phenomena covered by the implicit rolesets in UMR. Moreover, even if such structures are identified, the use of the relevant implicit roleset typically implies a different structure. Compare, for example, the lower part of (2), with *každý rok* 'every year' specifying frequency; the use of the rate-entity-91 roleset with its :ARG3 role (together with the abstract entity temporal-quantity, see § 4.3 below) makes the structure fairly different.

In particular, abstract predicates indicating meta-language information and those related to dialog structures represent a significant source of differences between the manual annotations and the automatic conversions. As this information is typically not explicitly structured by the language, it is not captured within the deep syntactic annotation (our source data), and thus cannot be straightforwardly converted. This is especially relevant for PDTSC dialogs, as illustrated in (3). The manual UMR structure clearly identifies the speaker and the listener and their role changing through the coreference annotation, in contrast to PDT (and thus to the automatic conversion).

(3) a. *Byla to vaše první motorka?*
'Was this your first motorcycle?'
b. *První.*
'First.'

```
MAN:
(s1s / say-91
  :ARG0 (s1e1 / person :refer-person 1st)
  :ARG2 (s1e2 / person :refer-person 2nd)
  :ARG1 (s1b / have-ord-91
          :quote s1s
          :ARG1 (s1m / motorka  'motorcycle'
                  :ARG1-of (s1b2 / belong-91
                             :ARG2 s1e2)) 'you'
          :ARG2 (s1p / ordinal-entity :value 1)))

(s2s / say-91
  :ARG0 (s2e1 / person :refer-person 1st)
  :ARG2 (s2e2 / person :refer-person 2nd)
  :ARG1 (s2b / have-ord-91
          :quote s2s
          :ARG1 (s2m / motorka  'motorcycle'
                  :ARG1-of (s1b2 / belong-91
                             :ARG2 s2e1)) 'I'
```

[11] A possible way to eliminate this type of disagreement would be to normalize all graphs into reified forms prior to an automatic evaluation.

```
        :ARG2 (s2p / ordinal-entity :value 1)))
:coref ((s1e1 :same-entity s2e2)
        (s1e2 :same-entity s2e1)
        (s1m :same-entity s2m))

AUTO:
(s1b / být-007  'be'
  :ARG1 (s1t / ten)
  :ARG2 (s1m / motorka  'motorcycle'
     :mod (s1e2 / entity :refer-person 2nd) 'you'
     :mod (s1p / první)))  'first'

(s2m / motorka  'motorcycle'
     :mod (s2p / první))  'first'
```

This example illustrates one more open question in the UMR specification: To what extent should UMR annotation reconstruct fragmentary usages and ellipses (highly relevant especially for spoken data and dialogs)? While the complete replay is reconstructed in the manual annotation (= *Motorka to byla moje první.* 'This was my first motorcycle.'), the PDT annotation, and thus the conversion, is limited to the fragment (= *První motorka.* 'The first motorcycle.')

Next steps: Although not explicitly annotated in our source files, meta-language information is also available within the PDT data. The next step, therefore, is to examine the extent to which UMR-relevant data can be extracted and utilized to enhance the conversion process: not only to identify speakers in spoken data but also to recognize elements such as headlines and other pertinent contextual information. Second, more detailed guidelines on proper UMR annotation of fragmentary sentences would improve data consistency.

4.2 Event-related nouns

The UMR specification suggests representing agent nouns as arguments of the respective verbs; thus, for example, *teacher* is a person annotated as :ARG0 participant of the predicate *teach-01*. One might infer that nouns denoting other participants should also be represented with respective eventive concepts (e.g., *food* can be annotated either as a thing being :ARG1 of *eat-01* or just as an instance of the lexical entity *food*). However, it is not clear how far the abstraction should go.

The possibility of multiple correct UMR structures for the same lexical content undermines the potential of any automatic metric considering just one "gold" annotation. It inevitably fails to provide comprehensive insight into the quality of the conversion. Cf. the following text fragment from the beginning of the Czech data (4).

(4) *Vážení čtenáři, …*
'Dear readers (= subscribers), …'

```
MAN:
(... :vocative (p / person
                  :ARG0-of (c / číst-002) 'read'
                  :mod (v / vážený))) 'dear'

AUTO:
(... :vocative (c / čtenář  'reader'
                   :mod (v / vážený))) 'dear'
```

Although both structures are correct UMRs, their proper comparison remains a challenge far exceeding the capabilities of a simple automatic metric.

Next steps: Though PDT-scenario does not distinguish which lexemes (words) are related to eventive concepts (verbal predicates) and which are entities, additional language resources can be used to identify at least unquestionable candidates for conversion (as already discussed in Lopatková et al., 2024). In addition, a more detailed specification of the UMR conventions could help reduce the occurrence of such ambiguous cases.

4.3 Abstract entities

Artificial lemmas employed in the PDT-scenario for unexpressed arguments (e.g., #PersPron, #EmpVerb) roughly correspond to UMR basic abstract concepts like person, thing, event. However, since it is not possible to deduce the correct type from PDT and LDT data automatically, the conversion introduces two supertypes: (i) entity, subsuming all UMR non-events (esp. person and thing), and (ii) concept (used esp. in constructions where two or more events, states, or entities are compared). The first supertype is illustrated in ex. (3), where the node s1e2 /person (the possessor) in the manual graph corresponds to the node s1e2 / entity in the converted one.

Further, UMR employs a rich set of abstract entities that identify structured data; for example:

- "entities" (e.g., url-entity, percentage-entity, or ordinal-entity in ex. (3), with the subrole :value),
- "quantities" (e.g., temporal-quantity *každý rok* 'every year' in ex. (2)),

In the current version of the conversion procedure, structured data of these types have not yet been processed using abstract entities. Thus, they represent an additional source of disagreement in our comparison. (Semi-)automatic identification of at least most frequent constructions remains one of the important tasks for further improvement.

4.4 Discourse relations

The PDT and UMR schemata represent paratactic structures (such as coordination and discourse relations) in a similar way, by introducing a dedicated node in the graph to represent the whole paratactic construction. As a result, the conversion process is generally straightforward and primarily concerned with technical adjustments. However, when paratactic constructions intertwine with other phenomena (such as relative clauses, represented in UMR as inverted relations) additional complexity arises, making the conversion less trivial. For instance, in example (5) (simplified), the conversion fails to accurately capture the :ARG0-of inverted relations, and the coordinating node and is incorrectly placed one level lower in the graph structure.

(5) *et qui fecere et qui facta aliorum scripsere, multi laudantur.*
'many who have acted, and many who have recorded the actions of others, are praised.'

```
MAN:
(sl / laudo-08   'praise'
   :ARG1 (a / and
      :op1 (p / person
         :quant (m / multus)   'many'
         :ARG0-of (f / facio-02   'act'          : ...))
      :op2 (p2 / person
         :quant m
       :ARG0-of (s / scribo-14   'write, record' : ...))))

AUTO:
(l / laudo-08   'praise'
   :ARG0 (e / entity)
   :ARG1 (m / multus   'many'
      :mod (a / and
         :op1 (f / facio-23   'act'          : ...)
         :op2 (s / scribo-14   'write, record'  : ...))))
```

Next steps: While discourse relations are generally handled correctly, their interaction with more complex constructions will be examined. Conversion will be refined if systematic errors are found.

5 Conclusions

This paper presents a comparison between manually constructed UMRs and those produced by automatic conversion from deep syntactic annotations in existing corpora——specifically, PDT-C for Czech and LDT for Latin. We employed a novel evaluation metric that offers several advantages over existing methods to assess similarity of UMR graphs. The results revealed limitations of the current conversion process, which we further analyzed to suggest areas of possible improvements.

Overall, our evaluation shows that automatic UMR conversion performs comparably for Czech and Latin. However, the analysis also reveals significant challenges inherent to the task, particularly the high level of semantic abstraction required by UMR and the fact that UMR allows for multiple valid representations with varying degrees of granularity. These characteristics complicate both the conversion itself and the evaluation of its accuracy.

Despite the relatively low scores, a simple visual comparison of manual and automatically created graphs often reveals reasonable alignment. This suggests that the automatic procedure—especially after implementing the proposed improvements—could serve as a solid basis for subsequent manual annotation, significantly accelerating and reducing the cost of creating UMR data.

Acknowledgments

The work described herein has been supported by the grants *Language Understanding: from Syntax to Discourse* of the Czech Science Foundation (Project No. 20-16819X) and *LINDAT/CLARIAH-CZ* (Project No. LM2023062) of the Ministry of Education, Youth, and Sports of the Czech Republic. It has also been partially supported by the Charles University, GAUK project No. 104924, and SVV project No. 260 821.

The project has been using data and tools provided by the *LINDAT/CLARIAH-CZ Research Infrastructure* (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2023062).

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING-ACL'98: Proceedings of the Conference*, pages 86–90, Montreal, Canada.

David Bamman and Gregory Crane. 2006. The design and use of a Latin dependency treebank. In *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT2006)*, pages 67–78. Citeseer.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Julia Bonn, Claire Bonial, Matt Buchholz, Hsiao-Jung Cheng, Alvin Chen, Ching-wen Chen, Andrew Cowell, William Croft, Lukas Denk, Ahmed Elsayed,

Eva Fučíková, Federica Gamba, Carlos Gomez, Jan Hajič, Eva Hajičová, Jiří Havelka, Loden Havenmeier, Ath Kilgore, Veronika Kolářová, and 40 others. 2025. Uniform meaning representation 2.0. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Julia Bonn, Matthew J. Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajič, Kenneth Lai, James H. Martin, Skatje Myers, Alexis Palmer, Martha Palmer, Claire Benet Post, James Pustejovsky, Kristine Stenzel, Haibo Sun, Zdeňka Urešová, Rosa Vallejos, and 4 others. 2024. Building a broad infrastructure for uniform meaning representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2537–2547, Torino, Italia. ELRA and ICCL.

Julia Bonn, Chen Ching-wen, James Andrew Cowell, William Croft, Lukas Denk, Jan Hajič, Kenneth Lai, Martha Palmer, Alexis Palmer, James Pustejovsky, Haibo Sun, Rosa Vallejos Yopán, Jens Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2023a. Uniform meaning representation. LINDAT/CLARIAH-CZ Digital Library, ÚFAL, MFF UK, Prague, Czechia.

Julia Bonn, Skatje Myers, Jens E. L. Van Gysel, Lukas Denk, Meagan Vigus, Jin Zhao, Andrew Cowell, William Croft, Jan Hajič, James H. Martin, Alexis Palmer, Martha Palmer, James Pustejovsky, Zdenka Urešová, Rosa Vallejos, and Nianwen Xue. 2023b. Mapping AMR to UMR: Resources for adapting existing corpora for cross-lingual compatibility. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 74–95, Washington, D.C. Association for Computational Linguistics.

Matthew J. Buchholz, Julia Bonn, Claire Benet Post, Andrew Cowell, and Alexis Palmer. 2024. Bootstrapping UMR annotations for Arapaho from language documentation resources. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2447–2457, Torino, Italia. ELRA and ICCL.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

William Croft and D. Alan Cruse. 2004. *Cognitive Linguistics*. Cambridge University Press, Cambridge.

Donald Davidson. 1967. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press.

Charles J. Fillmore, Collin Baker, and Hiroaki Sato. 2002. Seeing Arguments through Transparent Structures. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation, LREC 2002*, pages 787–791, Paris. ELRA.

Federica Gamba, Alexis Palmer, and Daniel Zeman. 2025. Boostrapping UMRs from Universal Dependencies for scalable multilingual annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX)*, Vienna, Austria. Association for Computational Linguistics.

Berta Gonzalez Saavedra and Marco Carlo Passarotti. 2014. Challenges in enhancing the Index Thomisticus treebank with semantic and pragmatic annotation. In *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT-13)*, pages 265–270.

Jan Hajič, Eduard Bejček, Alevtina Bémová, Eva Buráňová, Eva Fučíková, Eva Hajičová, Jiří Havelka, Jaroslava Hlaváčová, Petr Homola, Pavel Ircing, Jiří Kárník, Václava Kettnerová, Natalia Klyueva, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, David Mareček, Marie Mikulová, Jiří Mírovský, and 26 others. 2024a. Prague dependency treebank - consolidated 2.0 (PDT-C 2.0). LINDAT/CLARIAH-CZ Digital Library, ÚFAL, MFF UK, Prague, Czechia.

Jan Hajič, Eduard Bejček, Jaroslava Hlaváčová, Marie Mikulová, Milan Straka, Jan Štěpánek, and Barbora Štěpánková. 2020. Prague dependency treebank - consolidated 1.0. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, pages 5208–5218, Marseille, France. European Language Resources Association.

Jan Hajič, Eva Fučíková, Markéta Lopatková, and Zdeňka Urešová. 2024b. Mapping Czech Verbal Valency to PropBank Argument Labels. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations (DMR 2024)*, pages 88–100, Torino, Italia. ELRA and ICCL.

R. W. Langacker. 1987. *Foundations of Cognitive Grammar: Theoretical Prerequisites*, volume I. Stanford University Press, Stanford.

Markéta Lopatková, Eva Fučíková, Federica Gamba, Jan Hajič, Hana Hledíková, Marie Mikulová, Michal Novák, Jan Štěpánek, Daniel Zeman, and Šárka Zikánová. 2025. UMR 2.0 - Czech: Release Notes. Technical Report TR-2025-74, ÚFAL MFF UK, Prague, Czechia.

Markéta Lopatková, Eva Fučíková, Federica Gamba, Jan Štěpánek, Daniel Zeman, and Šárka Zikánová. 2024. Towards a conversion of the prague dependency treebank data to the uniform meaning representation. In *Proceedings of the 24th Conference Information Technologies – Applications and Theory (ITAT 2024)*, pages 62–76, Košice, Slovakia. Univerzita Pavla Jozefa Šafárika v Košiciach, Košice, Slovakia, CEUR-WS.org.

Marco Passarotti. 2014. From syntax to semantics. first steps towards tectogrammatical annotation of Latin. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 100–109, Gothenburg, Sweden. Association for Computational Linguistics.

Marco Passarotti. 2019. The Project of the Index Thomisticus Treebank. *Digital Classical Philology*, 10:299–320.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Reidel, Dordrecht.

Haibo Sun and Nianwen Xue. 2024. Anchor and broadcast: An efficient concept alignment approach for evaluation of semantic graphs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1052–1062, Torino, Italia. ELRA and ICCL.

Jens van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, James Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, and Rosa Vallejos. 2021. Designing a uniform meaning representation for natural language processing. *KI - Künstliche Intelligenz*, 35(2):343–360.

Shira Wein and Julia Bonn. 2023. Comparing UMR and cross-lingual adaptations of AMR. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations (DMR 2023)*, pages 23–33, Nancy, France. Association for Computational Linguistics.

A Node mapping in *ju:mætʃ* and *smatch*

Here we show an example sentence from the test data and document the word alignments and the node mapping used by the two metrics.

The full sentence: *Vážení čtenáři, je tomu právě rok, kdy jsme vám oznamovali nepopulární informaci, že se cena našich novin zvyšuje.* "Dear readers, it's been a year since we announced the unpopular news that the price of our newspaper was increasing."

Our excerpt: *Vážení čtenáři, je tomu právě rok, kdy jsme vám oznamovali informaci* "Dear readers, it's been a year since we announced the news"

```
MAN:
(s1p0 / publication-91
  :ARG3 (s1s1 / say-91
    :aspect activity
    :modal-strength full-affirmative
    :ARG0 (s1p1 / person
      :refer-number plural
      :refer-person 1st)
    :ARG2 (s1p2 / person
      :refer-number plural
      :refer-person 2nd
      :ARG0-of (s1c1 / číst-002 'read'
        :aspect habitual
        :modal-strength full-affirmative)
      :mod (s1v1 / vážený 'dear'))
  :ARG1 (s1h1 / have-temporal-91
    :aspect state
    :modal-strength full-affirmative
    :quote s1s1
    :vocative s1p2
    :ARG1 (s1o1 / oznamovat-002 'announce'
      :aspect performance
      :modal-strength full-affirmative
      :ARG0 s1p1
      :ARG1 (s1i1 / informace 'information'
        :refer-number singular)
      :ARG2 s1p2)
    :ARG2 (s1r1 / rok 'year'
      :refer-number singular
      :mod (s1p4 / právě 'just')))))

AUTO:
(s1b1 / být-011
  :aspect activity
  :vocative (s1c1 / čtenář 'reader'
    :refer-number plural
    :mod (s1v1 / vážený 'dear'))
  :ARG1 (s1t1 / ten
    :refer-number singular
    :temporal (s1p1 / právě 'just'))
  :duration (s1r1 / rok 'year'
    :refer-number singular
    :temporal-of (s1o1 / oznamovat-002 'announce'
      :aspect activity
      :ARG0 (s1p2 / person
        :refer-number plural
        :refer-person 1st)
      :ARG1 (s1i1 / informace 'information'
        :refer-number singular)
      :ARG2 s1c1)))
```

ju:mætʃ node mapping between MAN and AUTO (word alignment, if any, is shown in brackets after the concept):

```
s1p0 / publication-91 … UNMAPPED
s1s1 / say-91 … UNMAPPED
s1p1 / person ("našich") … s1p2 / person ("našich")
s1p2 / person ("čtenáři vám")
      … s1c1 / čtenář ("čtenáři vám")
s1c1 / číst-002 … UNMAPPED
s1v1 / vážený ("Vážení") … s1v1 / vážený ("Vážení")
s1h1 / have-temporal-91 ("je") … s1b1 / být-011 ("je")
s1o1 / oznamovat-002 ("tomu jsme oznamovali")
      … s1o1 / oznamovat-002 ("jsme oznamovali")
s1i1 / informace ("informaci")
      … s1i1 / informace ("informaci")
UNMAPPED … s1t1 / ten ("tomu")
s1r1 / rok ("rok kdy") … s1r1 / rok ("rok kdy")
s1p4 / právě ("právě") … s1p1 / právě ("právě")
```

smatch node mapping between MAN and AUTO (showing only differences from *ju:mætʃ* mapping):

```
s1s1 / say-91 … s1b1 / být-011 ("je")
s1h1 / have-temporal-91 ("je") … s1t1 / ten ("tomu")
```

In our excerpt, the only nodes left unmapped by

smatch are s1p0 and s1c1 from the MAN graph, because there are no nodes left available in the AUTO graph. There are two other nodes that are left unmapped by *juːmæʧ* but not by *smatch*: s1s1 in MAN and s1t1 in AUTO. The mapping that *smatch* found for these nodes has no semantic justification (but it will slightly increase F_1 score because both say-91 and být-011 have :aspect activity).

The Role of PropBank Sense IDs in AMR-to-text Generation and Text-to-AMR Parsing

Thu Hoang
Amherst College
thuhoang28@amherst.edu

Mina Yang
Amherst College
miyang27@amherst.edu

Shira Wein
Amherst College
swein@amherst.edu

Abstract

The graph-based semantic representation Abstract Meaning Representation (AMR) incorporates Proposition Bank (PropBank) sense IDs to indicate the senses of nodes in the graph and specify their associated arguments. While this contributes to the semantic information captured in an AMR graph, the utility of incorporating sense IDs into AMR graphs has not been analyzed from a technological perspective, i.e. how useful sense IDs are to generating text from AMRs and how accurately senses are induced by AMR parsers. In this work, we examine the effects of altering or removing the sense IDs in the AMR graphs, by perturbing the sense data passed to AMR-to-text generation models. Additionally, for text-to-AMR parsing, we quantitatively and qualitatively verify the accuracy of sense IDs produced from state-of-the-art models. Our investigation reveals that sense IDs do contribute a small amount to accurate AMR-to-text generation, meaning they enhance AMR technologies, but may be disregarded when their reliance prohibits multilingual corpus development.

1 Introduction

The Proposition Bank (PropBank; Palmer et al., 2005) is a corpus of semantic roles of verbs and their arguments, where each verb sense is assigned an ID.[1] In addition to verbs, PropBank also annotates semantic roles of select adjectives, prepositions, and multiword expressions (Pradhan et al., 2022); some of the verbs in PropBank are verb-particle constructions, where a combination of a verb and preposition have a specified unique meaning, such as "turn in" meaning to submit/hand in.

The graph-based semantic representation Abstract Meaning Representation (AMR; Banarescu et al., 2013) uses English PropBank frames to indicate the sense of each node in the graph and its associated arguments (as shown for `like-01` in Figure 1). While the sense IDs in AMR graphs provide relevant semantic information, this inclusion requires manually checking PropBank for each sense ID and presents challenges when trying to annotate AMR in languages other than English (if adequate PropBank frames do not exist for that language). Senses do not always correspond across languages (Padó, 2007; van der Plas et al., 2010), limiting the benefits of relying on English PropBank for non-English languages, and many low-resource languages do not have framesets available. Two extensions of AMR, Uniform Meaning Representation (UMR; Van Gysel et al., 2021) and WISeR (Widely Interpretable Semantic Representation; Feng et al., 2023), resolve this issue by incorporating a "Stage 0" frameset development phase for low-resource languages and eliminating senses from the representation entirely, respectively.

Text: Everyone likes strawberries in summer.
Parsed AMR:

```
(l / like-01
    :ARG0 (e / everyone)
    :ARG1 (s / strawberry)
    :time (s2 / summer))
```

Figure 1: Example sentence and its AMR graph. The dashed number (`-01`) is the PropBank sense ID specifying the intended meaning of the predicate `like`.

Thus, given the prohibitive nature of sense IDs in multilingual extensions of AMR, in this work, we examine the technical utility of maintaining sense IDs in AMRs. We investigate the extent to which AMR-to-text generation models and text-to-AMR parsing models accurately rely on sense IDs when producing either text or AMRs, respectively. Specifically, we examine how AMR-to-text generation models perform when the sense IDs are altered in the AMR graphs, and perform an analysis of the

[1]For example, `like-01` and `like-02` are two different senses of `like`, where `like-01` means *have affection towards, be fond of, enjoy (habitually)* while `like-02` means *would like, wish, want (polite)* (Palmer et al., 2005).

Proceedings of the 6th International Workshop on Designing Meaning Representations, pages 13–18

accuracy of sense ID prediction in text-to-AMR parsers. We alter sense IDs in the input AMRs by:

- removing the sense IDs,
- replacing them with sense IDs that do not correspond with real PropBank frames,
- changing each sense ID to a realistic different sense of the same verb,
- swapping each verb's sense ID with the most (and least frequent) sense ID in the AMR3.0 corpus, and
- swapping each verb-particle construction with a verb (and the reverse: swapping all verbs with verb-particle constructions where possible).

We then generate text from four state-of-the-art AMR-to-text generation models on versions of the AMR3.0 dataset (Knight et al., 2020) with these sense ID alterations.

Next, we set out to ascertain the accuracy of sense IDs parsed by state-of-the-art automatic text-to-AMR parsers, which is related to the task of word sense disambiguation. We do this by examining whether the sense IDs match among the verbs that appear in both the automatically parsed AMRs and their human-annotated gold references.

We find that, while AMR-to-text generation models exhibit only a small decrease in automatic metric scores from these perturbations (removals and changes), there is still a statistically significant decrease for all models across all automatic metrics. We also find that, impressively, even for less frequently appearing senses, text-to-AMR parsers perform sense induction highly accurately. These results suggest that sense IDs are a contributing factor in the success of AMR technologies, but may be disregarded when necessary to promote multilingual extensions of AMR.

2 Methods

Here, we outline the data we use for experimentation (Section 2.1), the methods for sense ID alteration in AMR-to-text generation (Section 2.2), the evaluation techniques and models for AMR-to-text generation (Section 2.3), and the evaluation techniques and models for text-to-AMR parsing (Section 2.4).

2.1 Data

The AMR3.0 dataset contains 59,255 sentences written in English (from sources such as news and online forums), along with their matching gold (human-annotated) AMR graphs. We use only the test split of AMR3.0 to produce the altered datasets and generate parsed outputs, but identify the highest and lowest frequency sense IDs for each verb across the entire AMR3.0 dataset.

2.2 Sense ID Alterations

We evaluate the quality of AMR-to-text generation output under various conditions. We remove the sense IDs in four ways to observe how different components of a sense, such as the dash, signal the presence a predicate. We perform substitutions based on the frequency and existence of each individual sense ID to understand the effect of the appearance of senses in the training data. Lastly, we alter the verb-particle constructions to observe the impact of the verb form on the generated sentence.

Removed. We test removing sense IDs from AMR graphs in four ways: (1) completely remove the sense IDs and the dash preceding them (e.g. `get-01` to `get`), (2) remove the sense IDs but keep the dash preceding them (e.g. `get-01` to `get-`), (3) change all the sense IDs to `0` (e.g. `get-01` to `get-0`) , and (4) change all the sense IDs to 00 (e.g. `get-01` to `get-00`). We hypothesize that the dash functions as a marker for sense IDs, and therefore keeping the dash may improve sense induction performance compared to completely removing it, by signaling to the model that the preceding word is a predicate.

Arbitrarily large. We inspect the impact of a large sense ID that does not exist in PropBank by changing all the sense IDs to arbitrarily large numbers, randomized between 50 and 100, given that no sense IDs above 50 appear in PropBank.

Realistic substitution. Next, we change each sense ID to a random, "realistic" sense ID. If the word has multiple senses in PropBank, we substitute the current sense with another PropBank sense of the same verb form. If there is only one sense (`-01`), we substitute in `-02`.

Highest frequency. Here, we change each sense ID to the sense ID that appears most frequently for each verb in the AMR3.0 dataset. In the case of a tie (i.e. more than one sense has the same frequency), the lower numbered sense is used (given that it was added to PropBank first).

Lowest frequency. Similarly, we change each sense ID to the sense ID that appears the fewest number of times in the entire AMR3.0 dataset. In the case of a tie, the higher valued number is used

	amrlib			SPRING			BiBL			AMRBART		
Datasets	BERT	BLEU	MET.	BERT	BLEU	MET.	BERT	BLEU	MET.	BERT	BLEU	MET.
Baseline	0.9523	0.3869	0.7119	0.9589	0.4181	0.7366	0.9642	0.4753	0.7695	0.9651	0.4815	0.7732
Removed (1)	0.9512	0.3778	0.7074	0.9581	0.4126	0.7355	0.9631	0.4678	0.7674	0.9611	0.4399	0.7572
Removed (2)	0.9514	0.3815	0.7097	0.9577	0.4115	0.7328	0.9630	0.4669	0.7660	0.9614	0.4423	0.7575
Removed (3)	0.9516	0.3807	0.7089	0.9446	*0.3531*	*0.6852*	0.9604	0.4531	0.7534	0.9564	0.4111	*0.7385*
Removed (4)	0.9517	0.3789	0.7101	0.9583	0.4134	0.7351	0.9635	0.4713	0.7677	0.9612	0.4373	0.7579
Arbitrarily Large	0.9509	0.3753	0.7068	**0.9584**	0.4125	**0.7366**	0.9624	0.4644	0.7634	0.9602	0.4286	0.7531
Realistic Substitution	0.9519	0.3806	0.7104	0.9578	0.4088	0.7319	0.9624	0.4667	0.7624	0.9611	0.4390	0.7557
Highest Frequency	**0.9521**	**0.3852**	0.7111	0.9583	**0.4150**	0.7343	**0.9639**	0.4729	0.7676	**0.9645**	**0.4761**	**0.7693**
Lowest Frequency	0.9520	0.3836	**0.7111**	0.9582	0.4123	0.7338	0.9637	**0.4758**	**0.7690**	0.9637	0.4711	0.7683
To VPC	*0.9348*	*0.3088*	0.6763	*0.9429*	0.3566	0.7041	*0.9495*	*0.4122*	*0.7430*	*0.9499*	*0.4056*	0.7425
Remove VPC	0.9407	0.3175	*0.6601*	0.9497	0.3785	0.7102	0.9575	0.4451	0.7554	0.9584	0.4469	0.7601

Table 1: AMR-to-text generation results on the baseline and ten altered versions (VPC=verb-particle construction). The highest non-baseline scores within each model are bolded in blue, and the lowest scores for each model are italicized in red.

(given that it was added to PropBank later).

Change to verb-particle construction. Where possible, we change each verb to a verb-particle construction, such as `get-away-08` or `run-out-05`. To test the significance in changing a verb to a verb-particle construction, we exclude all AMR graphs that did not have any changes made (i.e.: verb has no verb-particle construction in PropBank). If there are valid senses to substitute, we choose one randomly. For example, if `drop-05` appears in the dataset, we replace it with a randomly chosen sense from the list: `drop-by-02`, `drop-off-03`, `drop-out-04`, `drop-in-08`. In this way, the parse is changed to have verb-particle construction (i.e. both the text and sense ID in the concept change) where applicable, though the verb-particle construction does not appear in the original sentence.

Remove verb-particle constructions. Finally, we change each verb-particle construction to a verb form, if applicable, using the same process as for changing to verb-particle constructions.

2.3 Generation Models & Evaluation

For AMR-to-text generation, we leverage four models: amrlib[2], SPRING (Bevilacqua et al., 2021), AMRBART (Bai et al., 2022), and BiBL (Cheng et al., 2022). For evaluation, we use the test set from AMR3.0 (Knight et al., 2020), which contains 1,898 AMR graphs, as some of these models were trained on the training portion of the corpus.

To analyze the effect of modifying sense IDs, we alter each node in the AMR graphs in the specified manner and then generate text from each of these sets of altered AMRs, using the aforementioned four generation models. We also generate baseline outputs from the original test split to compare how well our modified outputs perform.

We evaluate the generated text with BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and BERTscore (Zhang et al., 2020).

2.4 Parsing Models & Evaluation

For text-to-AMR parsing, we assess the accuracy of the sense IDs included in automatically produced AMR graphs. We use five models: the BART-large fine-tuned model of amrlib, SPRING, AMRBART, BiBL, and LeakDistill (Vasylenko et al., 2023). For evaluation, we use the test set from AMR3.0. Specifically, we use the Consensus dataset, which contains 100 AMRs, which we chose due to its suitable size for manual qualitative analysis.

In order to perform a small-scale analysis of text-to-AMR parser accuracy for sense IDs, we use the aforementioned parsers to generate the 100 predicted AMRs for each model. Then, we check for matching verbs, and of those verbs, correct sense IDs. For example, if the gold annotation is `get-01` and the predicted sense is `get-02`, then we have a matching verb and a different sense ID.

3 Results

Table 1 shows the results of our experiments on the effect of altering sense IDs on AMR-to-text generation; Table 2 contains the results of our evaluation of the sense ID accuracy of text-to-AMR parsing models.

3.1 Generation Results

We find that the automatic metric scores are only slightly—though consistently—lower for texts gen-

[2]amrlib GitHub Repository

erated from the altered datasets. This includes cases where sense IDs were removed, a promising finding for extending AMR-to-text generation to languages with insufficient PropBank frames.

Interestingly, the impact is more pronounced for better-performing models, suggesting they may be utilizing the sense ID information to a greater degree. In particular, AMRBART is the best-performing model with a baseline BERTscore of 0.9651, but its modified outputs show an average decrease of 0.0053. On the other hand, amrlib, SPRING, and BiBL have baseline BERTscores of 0.9523, 0.9589, and 0.9642, respectively, but their modified outputs show decreases of only 0.0035, 0.0045, and 0.0033 on average.

The text generated from the AMR nodes swapped with their highest frequency sense IDs has the highest automatic metric scores overall, with BERTscore decreases of just 0.0002 to 0.0006 compared to the baseline. This supports our hypothesis that AMR-to-text generation models tend to prioritize generating PropBank sense IDs based on their frequency in the AMR3.0 corpus. Notably, the AMRs swapped with the least frequent senses also perform competitively, occasionally outperforming all the other altered datasets in BLEU and METEOR scores. The highest and lowest frequency substitutions are the only alterations which ensure that all sense IDs present in the AMRs actually exist in PropBank, suggesting that maintaining valid sense information (and the same verb form) leads to higher quality text generation.

In contrast, AMRs involving verb-particle construction substitutions result in the greatest performance drops overall, with an average BERTscore decrease of 0.0122 across all models. These are the only cases where the root verbs change entirely, indicating that such changes disrupt the performance of AMR-to-text generation models more than changes to sense IDs alone.

We also find that the way in which the sense IDs are removed has an impact on the generated text, where maintaining the dash preceding the sense ID or changing the sense ID to `00` improves model performance compared to removing them both completely. This suggests that models treat the dash as a predicate marker. Furthermore, using `00` preserves the familiar formatting of most sense IDs and aligns with its use as a placeholder for missing predicates (Banarescu et al., 2019).

Though on an item-level basis the decrease in BERTscore is minimal, we find that *all* perturbations result in a statistically significant decrease in BERTscore when compared via t-tests ($p \leq 0.05$). We perform paired t-tests comparing the baseline BERTscore values against all datasets, except for the verb-particle construction changes, for which we perform unpaired t-tests given that these datasets are smaller (since not all individual AMR graphs were able to have a verb-particle construction substitution for any nodes).[3] This suggests that AMR-to-text generation models are still sensitive to changes in verb senses.

Models	Matching Verbs	Sense Accuracy (%)	1-Sense Verbs (%)
amrlib	351	98.0%	51.6%
SPRING	349	98.0%	51.6%
BiBL	341	98.5%	52.8%
AMRBART	350	98.9%	51.7%
LeakDistill	343	98.3%	52.5%

Table 2: Text-to-AMR parsing results. Sense Accuracy refers to instances where not only the root verbs but also the associated sense IDs are predicted correctly. About half of these matching verbs for each model have only one sense, with the exact percent for each model indicated here with the "1-Sense Verbs" column.

3.2 Parsing Qualitative Analysis

We check the sense accuracy of verbs which appear in both the gold AMR and the system output. As seen in Table 2, all five text-to-AMR parsers—amrlib, SPRING, BiBL, AMRBART, and LeakDistill—demonstrate high accuracy in assigning sense IDs to correctly predicted verbs, with accuracy rates from 98.0% to 98.9%. About half of these matching verbs for each model have only one sense, contributing to this high accuracy.

Impressively, the parsers also correctly identify less frequent senses. For instance, all five models accurately predict `run-04` in a sentence about Route 288 in Virginia,[4] even though `run-04` appears only 19 times in the AMR3.0 training split—compared to 188 instances of `run-01` and 149 of `run-02`. However, one of those 19 instances mentions "Virginia_State_Route_203" in a similar context, suggesting that the models drew on contextual patterns from training.

Our study is conducted using the AMR3.0 corpus, which primarily consists of newswire and on-

[3] For amrlib, the p-values range from $<$0.0001 to 0.0317. For SPRING, the p-values range from $<$0.0001 to 0.0244. For BiBL, the p-values range from $<$0.0001 to 0.0050. Finally, for AMRBART, the p-values range from $<$0.0001 to 0.0047.

[4] "Route 288, the circumferential highway running around the south - western quadrant of the Richmond New Urban Region, opened in late 2004."

line text, raising the question of how our findings on sense ID sensitivity generalize to other domains. From our results, we find that text-to-AMR parsers perform sense induction accurately even for senses that appear infrequently in the training data. This is promising for applying parsing models to other corpora, such as *The Little Prince* dataset (Banarescu et al., 2013), which is a literary work with often uncommon language usage. Even if infrequent senses appear in other corpora, our findings suggest that the parsing models would still perform well. The relatively small decrease in generation quality from sense ID alterations suggests that generation models are not effectively using the sense ID information. It is unclear whether this is due to the model architecture or how the sense ID information appears in AMR graphs. However, we know that the presence of a dash improves performance, suggesting that models recognize this as a signal to expect sense IDs. Additionally, the substantial drop in performance when substituting for verb-particle construction indicates that the verb form has a larger impact than the sense ID itself.

4 Conclusion & Future Work

In this work, we explored to what degree AMR-to-text generation models rely on sense IDs in AMR graphs, by swapping or removing the sense IDs in the nodes, and assessing the quality of the resulting text. We find that AMR-to-text generation models are susceptible to sense perturbations and suffer a small decrease in automatic metric scores (BERTscore, BLEU, and METEOR), with BERTscore decreases of up to 0.0175; though the decrease is relatively small, all of the changes that we make to the sense IDs result in a statistically significant decrease in text quality for all generation models. We also measured the accuracy of sense annotation in text-to-AMR parsers, and our parsing analysis reveals that AMR technologies do accurately perform sense induction when parsing.

Our results indicate that sense IDs enable higher quality text generation when included in the AMRs for AMR-to-text generation models, and provide insightful semantic content within the AMR. Still, the technical relevance of sense IDs is small, and may be worth avoiding if the creation of in-language frames precludes the development a non-English AMR extension—or for multilingual extensions of AMR broadly. Accordingly, our findings motivate future work investigating multilingual extensions of AMR that do not include any sense IDs and generalize roles across (i.e. moving from opaque arguments such as `:ARG0` to more generalizable terms such as `:agent`); finding generic terms that would be sufficiently representative across languages presents an additional challenge.

Acknowledgments

Thank you to our reviewers and members of the Amherst College ACNLP lab for feedback on our draft. This work is supported by the Amherst College HPC, which is funded by NSF Award 2117377.

References

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2019. Abstract Meaning Representation (AMR) 1.2.6 specification. https://github.com/amrisi/amr-guidelines/blob/master/amr.md.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. In *Proceedings of AAAI*.

Ziming Cheng, Zuchao Li, and Hai Zhao. 2022. BiBL: AMR parsing and generation with bidirectional Bayesian learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5461–5475, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Lydia Feng, Gregor Williamson, Han He, and Jinho D. Choi. 2023. Widely interpretable semantic representation: Frameless meaning representation for broader applicability. *Preprint*, arXiv:2309.06460.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and 1 others. 2020. Abstract Meaning Representation (AMR) Annotation Release 3.0. Technical Report LDC2020T02, Linguistic Data Consortium, Philadelphia, PA.

Sebastian Padó. 2007. *Cross-lingual annotation projection models for role-semantic information*. Ph.D. thesis, Saarland University.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Sameer Pradhan, Julia Bonn, Skatje Myers, Kathryn Conger, Tim O'gorman, James Gung, Kristin Wrightbettner, and Martha Palmer. 2022. PropBank comes of Age—Larger, smarter, and more diverse. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 278–288, Seattle, Washington. Association for Computational Linguistics.

Lonneke van der Plas, Tanja Samardžić, and Paola Merlo. 2010. Cross-lingual validity of PropBank in the manual annotation of French. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 113–117, Uppsala, Sweden. Association for Computational Linguistics.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, and 1 others. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

Pavlo Vasylenko, Pere-Lluís Huguet Cabot, Abelardo Carlos Martínez Lorenzo, and Roberto Navigli. 2023. Incorporating graph information in transformer-based amr parsing. In *Findings of ACL*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *Proc. of ICLR*, Online.

Boosting a Semantic Parser Using Treebank Trees Automatically Annotated with Unscoped Logical Forms

Miles Frank and Lenhart Schubert
mfrank14@u.rochester.edu
schubert@cs.rochester.edu
Department of Computer Science, University of Rochester, Rochester, NY 14627 USA

Abstract

Deriving structured semantic representations from unrestricted text, in a format suitable for sound, explainable reasoning, is an important goal for achieving AGI. Consequently much effort has been invested in this goal, but the proposed representations fall short in various ways. Unscoped Logical Form (ULF) is a strictly typed, loss-free semantic representation close to surface form and conducive to linguistic inference. ULF can be further resolved into the more precise Episodic Logic. Previous transformer language models have shown promise in the task of parsing English to ULF, but suffered from a lack of a substantial dataset for training. We present a new fine-tuned language model parser for ULF, trained on a greatly expanded dataset of ULFs automatically derived from Brown corpus Treebank parse trees. Additionally, the model uses Parameter Efficient Fine Tuning (PEFT) to leverage a substantially larger base model than its predecessor while maintaining fast training times. We find that training on automatically derived ULFs substantially improves parser performance from the existing smaller dataset (from SEMBLEU score of 0.43 to 0.68), or even the previously used larger, generatively augmented ULF dataset, used with a transition parser (from SEMBLEU score of 0.49 to 0.68).

1 Introduction

Large language models (LLMs) have revolutionized the interactive generation of fluent, coherent text by machines, but their functioning is hidden in their millions or billions of parameters. This blurs the distinction between knowledgeable output and confabulation. Moreover, because they rely on probabilistic mimicry of their vast training data, rather than on rational thought, they do not reason or plan with the kind of reliability and scalability that is required for consequential applications in areas like healthcare, legal matters, police operations, or search and rescue. Ultimately, artificial general intelligence (AGI) requires the ability to reason and plan reliably at scale, and to explain how conclusions or plans were arrived at. For reasoning to be explicit and auditable, the knowledge and rules employed must themselves be made explicit and sufficiently unambiguous. You cannot tell whether *"Alice warned the woman that Bob had left"* plausibly entails *"Bob had left"* or instead, *"Bob had left the woman,"* without clarifying the semantic structure of the premise.[1] Thus effective representation of linguistic content and background knowledge forms the cornerstone of systems designed not only to converse fluently, but also to reason and plan reliably. Such representations should be derivable from language, and enable semantic inference, discourse processing, and explicit, explainable reasoning. Kim and Schubert (2019) describe Unscoped Logical Form (ULF), one such knowledge representation (with a lengthy prior history, e.g., Hwang and Schubert, 1994; Schubert and Hwang, 2000), as an alternative to other popular representations, because it preserves more of the semantic information of natural language while maintaining a strict type system supporting well-founded, natural inference.

Due to their retention of all sentential information and their coherent type structure, ULFs lend themselves to *natural logic*-like inference (Kim et al., 2021c,b), discourse inferences including clause-taking verbs, counterfactuals, questions, requests, and generalizations (Kim et al., 2019), as well as schema-based story representation (Lawley et al., 2019). ULFs, and their subsequent resolu-

[1] As a preview, the alternative VP logical forms are these (hinging on reifier `that` vs. relativizer `that.rel`):

```
((PAST warn.v) (the.d woman.n)
 (that (| Bob| ((PAST have.aux) (PERF leave.v)))))

((PAST warn.v) (the.d (n+preds woman.n (sub that.rel
    (| Bob| ((PAST have.aux) ((PERF leave.v) *h)))))))
```


Proceedings of the 6th International Workshop on Designing Meaning Representations, pages 19–29
August 4, 2025.

tion into Episodic Logic, have also proven to be a useful representation for inference within interactive natural language understanding systems (Kane et al., 2020, 2023). Improving the scope and accuracy of ULF parsers will enable generalization of such systems. To provide an initial idea of the form of ULFs and their application to inference, here are three simple examples of the ULFs for the sentences *"Bob pretended to be asleep"*, *"Alice often kids Bob"*, and *"I wish I had turned off the stove"*, along with some inferences derivable by the cited methods:

((| Bob| ((PAST pretend.v) (to (be.v asleep.a)))))
⇒ (| Bob| ((PAST be.v) (not asleep.a)))

(| Alice| frequently.adv-f ((PRES kid.v) | Bob|))
⇒ ((a.d person.n) sometimes.adv-f ((PRES tease.v) (a.d person.n)))

(I.pro ((PRES wish.v) (tht (I.pro ((cf have.aux-s) ((PERF turn_off.v) (the.d stove.n)))))))
⇒ (I.pro ((PAST do.aux-s) not.adv-s (turn_off.v (the.d stove.n))))

(Some syntactic explanations follow later.) Their similarity to surface form should enable the reader to understand the inferences. Unlike inferences by LLMs, such ULF-based inferences are explainable in detail, in this case in terms of the implications of "pretending to," from the plausible assumption that "Bob" and "Alice" are instances of persons, from the entailment "frequently" ⇒ "sometimes," from the approximate synonymy of "kid" and "tease" (as verbs), and (in the last example) from the properties of counterfactual entailment of the subjunctive form. Resolving ULFs into Episodic Logic (EL) involves systematic deindexing, scoping, and reference resolution processes, and this more precise representation enables a superset of FOL inferences as well as uncertain inferences, in conjunction with miscellaneous world and lexical knowledge, and with support from taxonomic, temporal, arithmetic, and other specialist subsystems (e.g., Schubert, 2014). If necessary, ULF can be further converted to Episodic Logic for more granular inference. Resolving ULFs into Episodic Logic (EL) involves systematic deindexing, scoping, and reference resolution processes, and this more precise representation enables a superset of FOL inferences as well as uncertain inferences, in conjunction with miscellaneous world and lexical knowledge, and with support from taxonomic, temporal, arithmetic, and other specialist subsystems (e.g., Schubert, 2014).

The main contributions of this paper are (1) the demonstration that a large corpus of syntactically annotated sentences from a wide spectrum of sources (the Brown corpus) can be rather reliably mapped to ULF – an English-like, highly expressive, coherently typed initial logical form previously shown to be suitable for inference; and (2) the ULF-annotated sentences thus obtained together with a small hand-annotated "gold" training set can be used to fine-tune an LLM for semantic parsing, obtaining a level of accuracy strikingly better than obtained by previous ULF parsers, and comparable to results obtained for other, less comprehensive semantic representations that used much larger hand-annotated training sets than our "gold" corpus.

In the remaining sections, we comment on related representations and prior ULF parsers (Section 2), our rule-based annotation of the Brown corpus Penn Treebank (Marcus et al., 1993) POS tags to obtain a greatly expanded ULF training set (Section 3), our models for fine-tuning and the success metrics (Section 4), and the results with our methods, comparing these to relevant previous semantic parsers (Section 5). We summarize and reiterate our results in the Conclusion (Section 6).

2 Related Work

2.1 Other Knowledge Representations

We briefly discuss the pros and cons of other contemporary knowledge representations including generic First Order Logic (FOL), Discourse Representation Theory (DRT), Abstract Meaning Representation (AMR), and Minimal Recursion Semantics (MRS). Perhaps the most simply formatted representation, FOL is easy to generate inferences from and expressive enough to represent the meaning of most simple, matter-of-fact sentences. Through the use of various syntactic and semantic maneuvers, FOL can also be adapted to sentences that involve more subtle subject matter. However, the required circumlocutions are apt to be awkward and remote from surface form. For example, they may require explicit quantification over possible worlds, or functionalizing of all predicates and quantifiers, and application of a "Holds" or "Is True" predicate to functionalized sentences (Schubert, 2015).

To address some pronoun resolution issues in the conversion of natural language to FOL, Kamp (1981) and Heim (1982) developed Discourse Rep-

resentation Theory. The nested structures in this theory contain free variables to be dynamically interpreted; but because Discourse Representation Theory is convertible to FOL, it shares the expressive limitations of the latter. (An extension of DRT allowing for mental states and attitudes, MS-DRT, seems not to have been deployed as yet in semantic parsing.)

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is less focused on echoing the syntax of sentences, instead striving to represent sentences of similar meaning but different wording as the same AMR graph structure. This is useful in detecting meaning similarity or equivalence, and reduces the need for inferences, such as a "collide" event occurred, given that "Bob was injured in a collision". However, AMR drops important aspects of meaning (such as tense, and the distinction between hypothetical events and real ones), and makes insufficient commitments about the semantic types of its constituents (such as modifiers and quantifiers) to be suitable for reliable inference (again see Schubert, 2015, where other representations are considered as well). The more recent multilingual Uniform Meaning Representation (UMR) (Van Gysel et al., 2021) extends AMR to include temporal and modal dependencies, but due to limited training corpora, the only available parsers use a pipeline approach by first parsing the AMR and then automatically converting to UMR (Chun and Xue, 20240815–20240815).

In view of the considerable attention that AMR has received in the research literature of the last decade, some quick comparisons of AMR and ULF structures can provide an intuitive idea of their characteristics and differences, particularly for readers unfamiliar with ULF. Consider the sentences

> 1. *The broadcast asserted that chemicals were dumped into the river.*
> 2. *The broadcast showed chemicals being dumped into the river.*

The AMR representations of these sentences are identical except for the respective event predicates {assert-02, show-01}:

```
(z0 / {assert-02, show-01}
    :ARG0 (z1 / broadcast
    :ARG1 (z2 / dump-01
              :ARG1 (z3 / chemical)
              :destination (z4 / river)))
```

Note the free variables, generally assumed to be existentially bound at the top level. For version (1), this roughly says that a broadcast z1 asserts an event z2 of dumping a chemical z3 into a river z4. Besides the neglect of tense, one issue is that a dumping event is implicitly assumed to exist, not allowing for a false assertion ("assert" should create an opaque context). Another is that "assert" should take a proposition, not an event, as object argument. (You can assert the Second Amendment, but not the Second World War.) The AMR representation works better for version (2), insofar as it's entirely possible that a broadcast might show a chemical dumping event.

The following are the quite distinct ULF interpretations automatically obtained for (1) and (2) (where the tags ∼1, ∼2, ... indicate positions of corresponding input words, needed for reference resolution and other pragmatic phenomena; they are omitted for ULF evaluations):

```
(((the.d~1 broadcast.n~2)
 ((PAST assert.v~3)
  (that~4
   ((k (plur chemical.n~5))
    ((PAST be.aux~6)
     ((pasv dump.v~7)
    (adv-a (into.p~8 (the.d~9 river.n~10)))))))))
\.)

(((the.d~1 broadcast.n~2)
 ((PAST show.v~3)
  ((k (plur chemical.n~4))
   ((PROG be.aux~5)
    ((pasv dump.v~6)
     (adv-a (into.p~7 (the.d~8 river.n~9))))))))
\.)
```

Some points to note in these examples (as well as the earlier introductory ones) are type/sortal distinctions indicated by dot-suffixes like .d (determiner), .n (nominal predicate), .v (verbal predicate), etc.; and the retention of tense, definite determiners, and plurals. 'plur' shifts a predicate true or false of single entities to a predicate true or false of sets of entities. The operator 'k' type-shifts a monadic predicate P to the abstract *kind* (k P) whose realizations satisfy P.[2] Most notably, the type-shifting operator 'that' in the first ULF maps a sentence meaning to a propositional individual (see Kim and Schubert, 2019). While the proposition exists, it need not be true and the entities it introduces need not exist – this is a matter of inference, for instance for a trustworthy report. In the second ULF, the verbal predicate 'show.v' is treated as taking an object (theme) – namely chemicals, and a predicate – namely, the property of being dumped

[2] But acting on a kind entails acting on an instance of the kind – here, an instance of the kind, chemicals.

into the river, as arguments. (Predicate arguments cannot be quantified over, and the logic remains first-order.)

Minimal Recursion Semantics (MRS) (Flickinger et al., 2012) shares some features with DRT and AMR, though it deals fully with restricted quantification, attitudes, and other phenomena. In its "native form" it uses ordinary predicate + arguments syntax, but assigns names (handles) to predications, using these as placeholders for embedded predications. However, the semantic representations seem under-determined in terms of type structure, and are somewhat hard to understand, because of the indirectness of the structural descriptions – use of handles to flatten the representation, span indices to indicate the scope of handles, and arguments of predicates that include, besides handles (sometimes undefined), various types of unbound variables that are presumably to be closed existentially with some appropriate scope. It is unclear if MRS is intended for reasoning, but we are not aware of recent work in that direction.

2.2 Previous ULF Parsers

Kim et al. (2021a) introduced an LSTM-based transition parser trained on a small, hand-annotated "gold" corpus of English-ULF pairs, achieving accuracy on par with early AMR parsers trained on much larger datasets. Gibson and Lawley (2022) later used a fine-tuned autoregressive language model on the same corpus and reported similar performance, showing that such models can perform well even with limited training data. Their model used the idea from (Mager et al., 2020) and (Bevilacqua et al., 2021) that the parsing task could be performed by seq2seq models similar to previous AMR-to-text models. Building on these, Juvekar et al. (2023) generated a much larger synthetic dataset using the gold data as seed sentences. Their method, grounded in ULF type constraints and linguistic patterns, created up to 116,112 English-ULF pairs, slightly improving upon (Kim et al., 2021a) (see Section 5).

Here, we present a new parser based on a large language model (LLM) trained on ULFs automatically derived from the Brown Treebank, containing about 50,000 sentences (20 words long on average) from many genres. Unlike the original gold corpus, which lacked longer and structurally complex sentences due to annotation costs, the Brown corpus provides broader structural and topical diversity. Whereas Gibson and Lawley used GPT-Too (Mager et al., 2020)[3], we apply Parameter Efficient Fine Tuning (PEFT) to a larger base model for improved performance with minimal training overhead.

3 Expanding the ULF Training Data Using the Penn Treebank Corpus

We now describe how we obtained ULF formulas from Brown corpus Penn Treebank (Marcus et al., 1993) syntax trees, for use in fine-tuning the Gemma-2B model (and also GTP-Too, for comparison). The idea behind use of the Brown corpus was that syntactic constituency trees roughly indicate the compositional semantic structure of sentences, and this should facilitate transduction into ULF. For example, a syntactic VP structure of form

(VP (VBD saw) (NP (DT the) (JJ white) (NN swan)))

(in the Penn Treebank format) can be regarded as indicating that the meaning of the verb phrase is obtained by applying the meaning of the past-tense verb "saw" to the meaning of the object noun phrase (NP). The result is a monadic predicate that can be applied to the meaning of an NP subject such as (NNP Bob) to obtain a sentence meaning. Similarly, the structure of the object NP suggests functional application of the determiner (DT) meaning and the adjective (JJ) meaning to the meaning of the nominal predicate, (NN swan).

3.1 Rule-based adjustments to the Treebank trees

However, there are some immediate adjustments that are needed to obtain a type-coherent structure. First, the past-tense component of (VBD saw) actually has sentence-level significance, placing the seeing-event (with the white swan as its object) in the past relative to the time of assertion. In ULF, (VBD saw) is split into a pair of semantic constituents, (PAST see.v), where "see.v" is an object-taking and subject-taking predicate, and PAST is an unscoped tense operator. Second, the structure of the object NP is insufficient to determine that the adjective should first be applied to the nominal predicate, forming the meaning of "white swan"; this modified nominal predicate is then operated upon by the determiner. In ULF, such determiner phrases are again unscoped semantic constituents. The resulting ULF phrase is thus

[3] "GPT-Too" appears in the title of this paper, referring to small, medium, and large versions of GPT-2 used by the authors for English generation from AMR.

((PAST see.v) (the.d ((MOD-N white.a) swan.n)));

this incorporates a third adjustment, namely conversion of the predicate "white.a" to a nominal-modifier via type-shifting operator MOD-N. This is needed if we take the (natural) view that "white" is lexicalized as a simple predicate (consider "Snow is white"), rather than as a predicate modifier like "fake".[4]

Thus, while syntactic constituency provides a rough indication of semantic structure, a variety of adjustment rules are needed to map Treebank trees to ULF. We use nearly 400 such rules, dealing with issues such as different uses of quotes, punctuation and brackets, inserting silent complementizers, regularizing complex quantifiers (such as "almost all" or "one out of six"), interpreting auxiliaries, distinguishing prepositional phrases used as predicates, predicate modifiers, or argument-suppliers, distinguishing the different semantic functions of participial VPs and subordinate clauses, expanding quantifying pronouns into quantifier-noun combinations (e.g., "nothing," "everybody"), dealing with displaced constituents, interpreting several types of comparatives, and many more.

The writing of these rules was made relatively straightforward by use of our tree transduction language TT, a simpler, more easily used variant of TTT (Purtee and Schubert, 2012). TT match patterns closely mirror the input tree structure, i.e., every sublist in a pattern must correspond to a sublist in the target list structure. The simplest pattern elements can be integers $i = 1, 2, ...$, which will match up to i successive atoms or lists. More often, we make use of TT's regex-like constructs, based on match predicates starting with characters '!', '?', '*', '+' to signal matchability to 1 item, 0 or 1 item, 0 or more items, and 1 or more items respectively; there are over 100 such predicates (separately defined). Some cover extensive data, for example, `!event-noun` covers about 220 event nouns, and a predicate checking for purely intransitive verbs covers over 5,300 verbs. A second class of match predicates, starting with a dot and applicable to atoms only, are interpreted via ISA-hierarchies. For example, `.TIME-PERIOD` checks whether the atom being matched "is a" word like *second, day, summer, pause, ...*, by checking for an ISA-chain of 0 or more links from the word to `.TIME-PERIOD`. (Lexical category can be checked by another ISA-predicate such as `.NN/NNP`, defined to match either `NN` or `NNP`.) Since TT allows for arbitrary nesting of expressions, the match predicates can be used at any structural level. Here is an example of the use of this language to expand a temporal NP such as "last summer," as represented in a constituent tree, into a temporal adverbial "during last summer":

```
(defrule *add-prep-for-definite-embedded-time-np*
 ; E.g., "I know what you did {last summer}"
 ;     parse fragment: (VP (AUX DID)
 ;                         (NP (JJ LAST) (NN SUMMER)))
 '((!atom *expr (!not-prep-or-symb +expr)
       (NP +expr (.NN/NNP .TIME-PERIOD)) *expr)
   (1 2 3 (ADVP (-SYMB- adv-e)
       (PP (-SYMB- {during}.p) 4)) 5)))
```

Every rule consists of a match pattern and an output pattern. Here the match pattern (!atom *expr (!not-prep-or-symb ...) (NP ...) *expr) matches any phrase in parentheses starting with exactly one atomic expression, followed by zero or more arbitrary expressions, followed by two subexpressions of specified forms (the second one being the temporal NP), and possibly additional ones.

When a match succeeds, the matched constituents can be referenced in the output pattern by their position. In the example, position indices 1–5 correspond to the five top-level matched expressions. Non-numeric elements are copied into the output directly, though TT also allows for output elements that are functions of matched input elements. Note the PP adverbial containing during.p (with the time-NP as its complement) in the output. To refer numerically to matched constituents lying within subexpressions of the match pattern, TT uses integers joined by dots. For example, 4.3.2 would refer to whatever piece of the input expression matched `.TIME-PERIOD`.

3.2 From adjusted trees to ULFs

Once transformed, trees are semantically interpreted via a compositional process driven by syntactic types and morphological cues. Lexemes receive type tags via about 50 rules based on word POS – which in many cases has been made semantically more revealing through preprocessing rules, e.g., WDT-REL instead of WDT for *which* or *that* used as a relativizer. Type-shifting operators introduced during preprocessing likewise facilitate function-argument application throughout. The compositional mapping from preprocessed phrases to ULFs is then quite simple, involving a little over

[4] Modified nominals cannot in general be viewed as a conjunction of two predicates, as in "is white and is a swan"; for instance this fails for "white wine," "plastic swan," or "utmost danger".

a page of code.

ULFs derived this way proved effective: in a small evaluation (11 sentences), the raw Brown-derived ULFs scored 0.81 F1 on EL-SMATCH and 0.82 on SEMBLEU, with 952 triples. Our final dataset includes 51,649 English-ULF pairs—substantially larger and more varied than the original gold corpus.

4 Models and Metrics

4.1 Language base models

Our model for deriving ULF from English builds on the training architecture developed by Gibson and Lawley (2022), which in turn built on GPT-Too, an AMR-to-English system (Mager et al., 2020). When run in reverse, Gibson and Lawley's model was shown to also be state-of-the art for the English to ULF parsing task. We apply Gibson and Lawley's architecture, fine-tuning on English-ULF sentence pairs to maximize the joint probabilities of English and ULF tokens. We also use their training process, but instead fine-tune Quantized Low Rank Adapters (QLoRA) (Dettmers et al., 2023) of the pretrained model to perform parameter-efficient fine-tuning (PEFT) to leverage a large base model. The previous LLM model used the 774M parameter version of GPT-Too (i.e., GPT-2L), while we use the 2.5B parameter Google Gemma-2B which would previously have been infeasible to train without parameter-efficient fine-tuning.

4.2 Metrics

We evaluated the model on both a test subset of the previous hand-annotated (gold) dataset ($n = 174$) and a test set of Brown corpus derived ULFs ($n = 174$) using the metrics EL-SMATCH and SEMBLEU. These metrics are borrowed from standard AMR evaluations, but the type-shifting operators of ULF and other differences from AMR require introduction of additional nodes and links to obtain Penman format, after which SMATCH and SEMBLEU can be applied. The SMATCH (Cai and Knight, 2013) score is calculated by (1) extracting all the triples from a hypothesis and reference AMR (e.g., see Figure 1), (2) performing a greedy search to unify variable names between the hypothesis and reference, and finally (3) calculating F1, precision, and recall scores from the matching triples. As noted by Groschwitz et al. (2023), current AMR parsers achieve high SMATCH scores but can still make frequent errors. This is partially because the SMATCH score suffers from two immediate problems: Only taking into account triples (two variables/concepts and a relations) means that larger semantic structure is not captured in the evaluation; and unifying the variables leads to over-counting matching triples where the relation matches but the variables do not map to the same concepts.

```
instance(z0, assert-02)     ARG0(z0, z1)
instance(z1, report-01)     ARG1(z0, z3)
instance(z2, news)          ARG1(z1, z2)
instance(z3, dump-01)       ARG1(z3, z4)
instance(z4, chemical)      destination(z3, z5)
instance(z5, river)
```

Figure 1: Extracted triples for the AMR corresponding to the sentence, "The news report asserted that chemicals were dumped into the river." `z0` through `z5` are variable names, the predicates `instance`, `ARG0`, `ARG1`, and `destination` are the edges of the AMR graph which capture semantic relations between variables. The `instance` predicate maps variables to concepts.

SEMBLEU scores are instead calculated by (1) extracting all n-grams from the hypothesis and reference AMR, where an n-gram includes n concepts connected by $n - 1$ relations (e.g., `assert-01 :ARG1 dump-01 :ARG1 chemical` is a 3-gram roughly corresponding to the meaning "chemicals being dumped is asserted"), (2) calculating an adjusted accuracy of matching n-grams between the hypothesis and reference, (3) multiplying by a brevity penalty. By including longer chains, SEMBLEU captures more complex semantic structures, and not using variables solves the over-counting problem of the SMATCH unification strategy. Because of this and in accordance with previous ULF parsing work, we use SEMBLEU (Song and Gildea, 2019) as a primary evaluation metric and EL-SMATCH for a more detailed breakdown of F1, precision and recall. EL-SMATCH is fully described by Kim and Schubert (2016), but is essentially an adaptation of SMATCH to evaluate ULFs as sets of triples in the same way as AMR.

5 Results

5.1 Results on the gold data in comparison with earlier ULF parsers

Using the 51,649 English-ULF dataset we obtained from the Brown corpus, and employing PEFT, we achieved major gains in all metrics as compared to previous ULF parsers – see Table 1. The results indicate that stronger base models improve evaluation metrics across the board, but have a less substantial effect than the new Brown-based dataset.

Base Model	SEMBLEU	EL-SMATCH		
		F1	Precision	Recall
(Kim et al., 2021a): Transition model	0.47	0.59		
(Gibson and Lawley, 2022): GPT-Too	0.43	0.63		
Trained on Gold + Generated Set				
(Juvekar et al., 2023): Transition model	0.49	0.60		
Trained on Gold + Brown Set (our results)				
GPT-2 124M	0.55	0.60	0.60	0.61
GPT-2 355M	0.66	0.69	0.70	0.68
Google Gemma 2B (PEFT)	**0.68**	**0.72**	**0.73**	**0.71**

Table 1: Results for models tuned on gold training set vs combined gold and Brown-derived training set.

The small gold dataset sufficed to train both Kim et al.'s transition-based and Gibson and Lawley's LLM-based ULF parser to a level of performance comparable with that of early AMR parsers trained on much larger datasets. As noted in Section 2, Juvekar et al. (2023) obtained small improvements over the original transition-based model using up to 116,112 artificially generated, type-consistent English-ULF pairs. The 51,649 English-ULF dataset we obtained from the Brown corpus is not as large as theirs, but we see substantial parsing performance increases over their parser. We suspect that this can be largely attributed to the fact that Brown Treebank sentences are a diverse, naturally occurring set, and that the carefully tuned, rule-based tree-to-ULF parser is almost as accurate as hand annotation of English sentences with ULFs. The substantial gains in SEMBLEU scores show that the model retrieves more individual constituents, and that the overall coherence of the fragments is higher.

5.2 Results on Brown-Derived ULFs

Our model's performance is best described by the results on the hand-annotated gold data. However, since our parser was fine-tuned on a combination of a (small) gold training set and a large set derived from the Brown corpus, it is of interest to look at its performance on Brown data in comparison with its performance on the gold data. Differences are to be expected, in part because the Brown data, though less accurate, clearly impacted performance very significantly, but also because some streamlining of certain syntactic conventions (e.g., the handling of auxiliary verbs and tense/aspect operators) was incorporated into the Brown data which are still in their old form in the gold data. The comparison is provided in Table 2.

As expected, the scores on the Brown-derived test set show substantially better SEMBLEU scores, although surprisingly, the EL-SMATCH scores are scarcely different. In other words, the parser generally matches the overall structure of Brown-derived data better than for gold data, perhaps because of the change in some ULF conventions, but the triple-by-triple match structure is not greatly affected. If we were to create a new gold set abiding by the revised conventions, our parser's performance likely would fall somewhere between the results on the gold and Brown-derived ULFs (i.e., between 0.68 and 0.76 on SEMBLEU). These results are also surprising because the sentence complexity and lengths in the Brown corpus are larger than those in the gold ULF set.

5.3 Comparison to AMR parsers

To relate our work to AMR parsing, we compare our ULF parsing results with results from two AMR parsers in Table 3. Other AMR parsers achieve similar SMATCH scores to (Drozdov et al., 2022) on the AMR 3.0 benchmark dataset. After the proof-of-concept GPT-Too parser (Mager et al., 2020), the first seq2seq parser with benchmark results (Bevilacqua et al., 2021), scored 83.0 on AMR 3.0. More recently Bai et al. (2022) and Vasylenko et al. (2023) build on (Bevilacqua et al., 2021) achieving significant improvements (scores of 84.2 and 84.6 respectively), using novel ideas such as incrementally finding spans to abstract, and inserting the corresponding concepts, treating the transduction between text and AMR as symmetric, and pretraining on AMR graph data rather than (just) text. For parsers of other knowledge representations, the recent English Resource Grammar parser by Lin et al. (2023) (based on Minimal Recursion Semantics) improves performance with a

Model	SEMBLEU	EL-SMATCH		
		F1	Precision	Recall
Gold ULF Test Set	0.68	0.72	**0.73**	0.71
Brown-Derived ULF Test Set	**0.76**	**0.72**	0.72	**0.72**

Table 2: Parser performance on hand-annotated (gold) test set versus performance on a test set of Brown-derived English-ULF pairs.

Parser Model	SEMBLEU	SMATCH/EL-SMATCH
AMR3-structbart-L (Drozdov et al., 2022)	0.56	0.83
AMR2-joint-ontowiki-seed42 (Lee et al., 2022)	0.60	**0.86**
Our Model	**0.68**	0.72

Table 3: Hand annotated test set comparison to AMR parser performance.

neural-symbolic approach, where prior knowledge from the symbolic parser alleviates inaccuracies of the neural model on out-of-distribution evaluation. A recent DRT parser from Yang et al. (2024) similarly proposes a neural-symbolic parser that predicts the scope structure with a rule or dependency based resolver.

As was seen in the discussion of sentences (1) and (2), the greater expressivity of ULF, and its fidelity to the full contents of sentences, results in more variety and complexity in ULF constructions relative to AMR. To re-emphasize this point, sentences such as "Dogs are barking" (thus, presently), "Dogs bark" (thus, generically), and "A dog barked" (thus, in the past) map to distinct ULF representations, while they are assigned the same AMR. This results in higher SMATCH scores for AMR parsers. Other knowledge representations also tend to blur semantic distinctions, or degrade for complex sentences (though apparently not for MRS). For example, DRT parsers score lower on datasets with long and complex sentences (SMATCH score of 87.1 on short example sentences versus 48.7 on longer sentences) (Yang et al., 2024).

Unlike the impressive SMATCH scores of AMR parsers, their SEMBLEU scores are weaker, suggesting that while they are able to adequately generate correct constituents, the arrangement of those constituents is less predictable than for ULF. While the greater expressivity and semantic fidelity of ULF may make it more difficult to generate individually correct constituents, the type coherence of ULF may also help improve the overall structure of the parses. When introducing the SEMBLEU evaluation metric, Song and Gildea (2019) show that SMATCH marks edges as identical regardless of the nodes they attach, leading to inflated scores for parsers that don't accurately capture sentence structure. From our increased SEMBLEU score, we tentatively infer that the ULF type structure is less susceptible to mistakes of this sort.

5.4 Error Analysis

The most common errors we observed in the results for testing on the gold test set were missing implicit references, not generating multi-sentence constructions, and incorrectly identifying proper nouns and quotations. Implicit references (semantic constituents not appearing in the surface text) should show up in ULFs as pronouns or other elements in curly brackets. Errors are possibly due to the Brown-derived ULFs having different proportions of the most common implicit references. The most common form in the gold ULFs is {YOU}.PRO (typically implicit in English imperatives), accounting for over half the implicit references in the gold test set but only 15% of the Brown-derived set. The latter contains more instances of {REF}.N and {FOR}.P (as in "This _ will serve _ to appease him," where the missing items are a nominal and a purposive "for" applied to the action type "to appease him"). Additionally, errors in multi-sentence constructions were expected because the Brown-derived ULFs only contain single sentence examples while the gold set contains examples with multiple punctuation-separated sentences.

The less frequent remaining errors include over-generating special operators and macros, and incorrect bracketing. Specifically, the parser over-generates the N+PREDS macro (typically used for combining a noun with its postmodifiers) which is again over-represented in the Brown-derived ULFs as compared to gold. Also the order in which pre- and post-modifiers are applied to a noun may

be different in gold sentence ULFs and in parser-generated ULFs, though it's sometimes unclear which order is correct. For example, the sentence "Name the disposable razor that 'costs about 19 cents.' " was hand annotated with

```
({you}.pro (name.v (the.d (n+preds
 ((mod-n disposable.a) razor.n)
  (that.rel ((PRES cost.v) (about.adv-s
   (ds currency ``19 cents''))))))))
```

but our model parses it to

```
({you}.pro (name.v (the.d
 ((mod-n disposable.a) (n+preds razor.n
  (that.rel ((PRES cost.v) ((about.mod-a | 19.a|)
   (plur cent.n)))))))))
```

These variant modifier structures have slightly different semantics, but neither is outright mistaken. The other difference between the hand annotation and the parse is the use of the domain-specific representation of currency in the gold ULF, (ds currency "19 cents") and the adv-s vs. mod-a difference. The Brown-derived ULFs do not include domain-specific annotations, so, naturally, the parser handles "19 cents" differently. Now, "19" to be suffixed with .a (the adjectival version of the numeral) and "about" is suffixed with .mod-a, so that it functions as an adjective modifier. In the hand-annotated sentence, the full "19 cents" is annotated in the domain-specific currency context, so there is no adjective 19.a for "about" to modify, and it is instead annotated with suffix .adv-s. Our model parses sentences like this well, but because of similar discrepancies that lead to larger differences from the hand-annotated ULF, their correctness is not reflected in our evaluation metrics.

6 Conclusion

We presented an LLM-based parser that demonstrates significant gains in parsing English to ULF, driven by a new dataset of English-ULF pairs automatically generated from Brown corpus Penn Treebank trees. These gains are evident across all metrics, especially SEMBLEU, which reflect the parser's ability to capture semantic relations and maintain coherence. Our approach outperforms previous ULF parsers and some modern AMR parsers, showing ULF's potential to represent nuanced semantics and complex sentence structures. While evaluation scores on gold test data are lower than on Brown-derived test data, this likely results from updates to ULF annotation principles since the gold data was created, so revising the gold data to align with current standards would be valuable.

With the new Brown ULF dataset, data scarcity is no longer the main challenge in ULF parsing. Future research can instead focus on incorporating learning techniques from AMR parsing, extending the augmentation strategy of Juvekar et al. (2023), or using ULF's type system to constrain generation.

The increased reliability of ULF parsing will make inference and reasoning in AI systems more broadly applicable. An example of a system that relied on rule-based semantic parsing into ULF was the DAVID virtual human (Kane et al., 2020) designed to answer questions in a physical "blocks world". DAVID was answered user questions like *"How many red blocks were to the left of a blue block, before I moved the Nvidia block?"*, based on observing and modeling blocks' spatial relations via cameras, and mapping questions to ULF for spatial model queries. Similarly, the SOPHIE system (Kane et al., 2023), a virtual cancer patient used to help train physicians, makes use of ULF inference in generating dialogue responses. The authors describe a future improvement to their system using a learned ULF parser, to support more logically coherent inferences within the global context.

An intriguing future research direction compatible with our approach to logical form would be to use the type structure of ULF for unsupervised language learning. It appears that the types of ULF and Episodic Logic—names, generalized quantifiers, predicates, predicate and sentence reifying operators, predicate and sentence modifying operators, and a handful more—suffice for human languages in general. We could treat these types as semantically "innate," and take language learning to be learning a mapping from word sequences to structures instantiating these types. The variability of languages, besides different vocabularies, would correspond to different strategies for linearizing and abbreviating internal graph-like structures to facilitate interpretation. Additional learning support besides textual corpora would be needed, such as visual grounding; but it seems that ULF/EL-like presupposed type structure should greatly reduce the demand for data in the learning process.

Acknowledgments

This research was sponsored in part by the University of Rochester's Schwartz Discover Grant. The authors are grateful for the guidance provided by Gene Kim and Lane Lawley. The referees' insights also enabled improvements to the paper.

References

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Jayeol Chun and Nianwen Xue. 20240815–20240815. Uniform meaning representation parsing as a pipelined approach. In *TextGraphs-17*, page 40–52.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. *arXiv Preprint arXiv:2305.14314 [cs.LG]*.

Andrew Drozdov, Jiawei Zhou, Radu Florian, Andrew McCallum, Tahira Naseem, Yoon Kim, and Ramon Fernandez Astudillo. 2022. Inducing and using alignments for transition-based AMR parsing. *arXiv Preprint arXiv:2205.01464 [cs.CL]*.

Dan Flickinger, Yi Zhang, and Valia Kordoni. 2012. DeepBank: a dynamically annotated treebank of the Wall Street Journal. In *Proc. of the 11th Int. Workshop on Treebanks and Linguistic Theories (TLT11)*, pages 85–96, Lisbon.

Erin Gibson and Lane Lawley. 2022. Language-model-based parsing and english generation for unscoped episodic logical forms. *The International FLAIRS Conference Proceedings*, 35.

Jonas Groschwitz, Shay Cohen, Lucia Donatelli, and Meaghan Fowlie. 2023. AMR parsing is far from solved: GrAPES, the granular AMR parsing evaluation suite. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10728–10752, Singapore. Association for Computational Linguistics.

Irene Heim. 1982. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, UMass Amherst.

Chung Hee Hwang and Lenhart K. Schubert. 1994. Meeting the interlocking needs of LF-computation, deindexing, and inference: An organic approach to general NLU. In *Proc. of the AAAI Fall Symposium, TR FS-94-04*, pages 1297–1302, New Orleans, LA.

Mandar Juvekar, Gene Kim, and Lenhart Schubert. 2023. Semantically informed data augmentation for unscoped episodic logical forms. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 116–133, Nancy, France. Association for Computational Linguistics.

Hans Kamp. 1981. A theory of truth and semantic representation. In P. Portner and B. H. Partee, editors, *Formal Semantics - the Essential Readings*, pages 189–222. Blackwell.

Benjamin Kane, Catherine Giugno, Lenhart Schubert, Kurtis Haut, Caleb Wohn, and Ehsan Hoque. 2023. Managing emotional dialogue for a virtual cancer patient: A schema-guided approach. *IEEE Transactions on Affective Computing, PrePrints*, pages 1–12.

Benjamin Kane, Georgiy Platonov, and Lenhart K. Schubert. 2020. Registering historical context in a spoken dialogue system for spatial question answering in a physical blocks world. In *Proc. of the 23rd Int. Conf. on Text, Speech and Dialogue (TSD 2020)*, pages 487–494, Brno, Czech Republic.

Gene Kim, Viet Duong, Xin Lu, and Lenhart Schubert. 2021a. A transition-based parser for unscoped episodic logical forms. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 184–201, Groningen, The Netherlands (online). Association for Computational Linguistics.

Gene Kim, Mandar Juvekar, Junis Ekmekciu, Viet Duong, and Lenhart Schubert. 2021b. A (mostly) symbolic system for monotonic inference with unscoped episodic logical forms. In *Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA)*, pages 71–80, Groningen, the Netherlands (online). Association for Computational Linguistics.

Gene Kim, Mandar Juvekar, and Lenhart Schubert. 2021c. Monotonic inference for underspecified episodic logic. In *Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA)*, pages 26–40, Groningen, the Netherlands (online). Association for Computational Linguistics.

Gene Kim, Benjamin Kane, Viet Duong, Muskaan Mendiratta, Graeme McGuire, Sophie Sackstein, Georgiy Platonov, and Lenhart Schubert. 2019. Generating discourse inferences from unscoped episodic logical formulas. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 56–65, Florence, Italy. Association for Computational Linguistics.

Gene Kim and Lenhart Schubert. 2016. High-fidelity lexical axiom construction from verb glosses. In

Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics, pages 34–44, Berlin, Germany. Association for Computational Linguistics.

Gene Louis Kim and Lenhart Schubert. 2019. A type-coherent, expressive representation as an initial step to language understanding. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 13–30, Gothenburg, Sweden. Association for Computational Linguistics.

Lane Lawley, Gene Louis Kim, and Lenhart Schubert. 2019. Towards natural language story understanding with rich logical schemas. In *Proceedings of the Sixth Workshop on Natural Language and Computer Science*, pages 11–22, Gothenburg, Sweden. Association for Computational Linguistics.

Young-Suk Lee, Ramón Astudillo, Hoang Thanh Lam, Tahira Naseem, Radu Florian, and Salim Roukos. 2022. Maximum Bayes Smatch ensemble distillation for AMR parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5379–5392, Seattle, United States. Association for Computational Linguistics.

Zi Lin, Jeremiah Liu, and Jingbo Shang. 2023. Neural-symbolic inference for robust autoregressive graph parsing via compositional uncertainty quantification. *Preprint*, arXiv:2301.11459.

Manuel Mager, Ramon Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020. GPT-too: A language-model-first approach for AMR-to-text generation. *arXiv Preprint arXiv:2005.09123 [cs.CL]*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the Penn Treebank. *Comput. Linguist.*, 19(2):313–330.

A. Purtee and L.K. Schubert. 2012. TTT: A tree transduction language for syntactic and semantic processing. In *EACL Workshop on Applications of Tree Automata Techniques in Natural Language Processing (ATANLP 2012)*, Avignon, France.

Lenhart Schubert. 2014. NLog-like inference and commonsense reasoning. *Linguistic Issues in Language Technology*, 9.

Lenhart Schubert. 2015. Semantic representation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 4132–4138. AAAI Press.

Lenhart K. Schubert and Chung Hee Hwang. 2000. Episodic Logic meets Little Red Riding Hood: A comprehensive natural representation for language understanding. In Lucja M. Iwańska and Stuart C. Shapiro, editors, *Natural Language Processing and Knowledge Representation*, pages 111–174. MIT Press, Cambridge, MA, USA.

Linfeng Song and Daniel Gildea. 2019. SemBleu: A robust metric for AMR parsing evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4547–4552, Florence, Italy. Association for Computational Linguistics.

Jens E. Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a uniform meaning representation for natural language processing. *KI - Künstliche Intelligenz*, 35:343–660.

Pavlo Vasylenko, Pere-Lluís Huguet Cabot, Abelardo Carlos Martínez Lorenzo, and Roberto Navigli. 2023. Incorporating graph information in transformer-based amr parsing. *Preprint*, arXiv:2306.13467.

Xiulin Yang, Jonas Groschwitz, Alexander Koller, and Johan Bos. 2024. Scope-enhanced compositional semantic parsing for drt. *Preprint*, arXiv:2407.01899.

Using MRS for Semantic Representation in Task-Oriented Dialogue

Denson George and Baber Khalid and Matthew Stone
Rutgers University
Piscataway, NJ 08854
firstname.lastname@rutgers.edu

Abstract

Task-oriented dialogue (TOD) requires capabilities such as lookahead planning, reasoning, and belief state tracking, which continue to present challenges for end-to-end methods based on large language models (LLMs). As a possible method of addressing these concerns, we are exploring the integration of structured semantic representations with planning inferences. As a first step in this project, we describe an algorithm for generating Minimal Recursion Semantics (MRS) from dependency parses, obtained from a machine learning (ML) syntactic parser, and validate its performance on a challenging cooking domain. Specifically, we compare predicate-argument relations recovered by our approach with predicate-argument relations annotated using Abstract Meaning Representation (AMR). Our system is consistent with the gold standard in 94.1% of relations.

1 Introduction

Natural Language Understanding (NLU) is a core capability of all dialogue systems. It enables machines to interpret and generate contextually appropriate responses to language. Semantic parsing has long been a crucial component of NLU, providing an early-stage component for converting language into a structured semantic representation. However, since the emergence of large language models (LLMs), there has been a trend towards entirely replacing NLU modules and structured semantic representations with end-to-end model inference (OpenAI, 2022). Such systems have been shown to perform well in question answering, natural language generation (NLG), translation, summarization, and many other applications (OpenAI, 2024). Nevertheless, state-of-the-art Task-Oriented Dialogue (TOD) systems still benefit from an NLU module or a semantic representation (Feng et al., 2021; Zhu et al., 2023; Sun et al., 2023), and outperform single-call LLM systems in specific TOD benchmarks (Hudeček and Dusek, 2023). LLMs struggle with key aspects of TOD, including lookahead planning problems (Bachmann and Nagarajan, 2024), reasoning (Jiang et al., 2024), and tracking belief states (Chiu et al., 2023). These issues highlight the potential advantages of having a structured semantic representation that can be updated based on dialogue, information from the environment, and plan-based task reasoning (Geib et al., 2022).

In this paper, we explore MRS as a semantic representation framework due to its rich expressive power, connections to logical inference, close links to syntax, and potential for constraint-based disambiguation (Copestake et al., 2005). We develop methods for benchmarking MRS approaches for dialogue based on annotations expressed in terms of Abstract Meaning Representations (AMR), by comparing the consistency of predicate-argument relations across representations, thus showing that MRS shows promise for TOD. Our evaluation shows that in 94.1% of cases, our implementation of MRS using spaCy yields edges consistent with gold-standard predicate-argument relations annotated in a cooking domain (Jiang et al., 2022).

2 Related Work

2.1 LLM TOD systems

Multiple recent TOD systems have been built using LLMs and specialized NLU modules for their specific task. However, most end-to-end LLMs can struggle in three areas. The first is with lookahead planning problems, where understanding the final goal is crucial to avoid early errors that can obstruct later steps. Bachmann and Nagarajan (2024) demonstrate cases where models trained to solve problems using only next-token prediction struggle to learn what the model should choose for the first token. Momennejad et al. (2023) and Valmeekam et al. (2023) found that models struggle on planning tasks framed as word problems. The second area

Proceedings of the 6th International Workshop on Designing Meaning Representations, pages 30–37

where LLMs may struggle is reasoning. Jiang et al. (2024) determined that state-of-the-art LLMs fail to reason consistently across minor variations, such as changing names of people or places. The third area is belief state tracking, where it has been seen that an end-to-end LLM inference compares poorly to supervised models. Hudeček and Dusek (2023) shows five state-of-the-art models performing better than LLMs, 3 of which use an NLU component or a semantic representation (Feng et al., 2021; Sun et al., 2023; Zhu et al., 2023).

LLM-based systems can stage multiple prompts to perform dialogue state tracking, knowledge retrieval, and dialogue planning (Dong et al., 2025; Xu et al., 2024; Zhang et al., 2023). However, as the amount of LLM calls or tokens in the output increase, the inference latency of LLMs can become a pain point for real-time dialogue systems; many AI assistants require a response within a particular time frame, such as Alexa's 8-second requirement for responses.[1] The specialized components that TOD systems use to achieve real-time performance—track belief states (Hudeček and Dusek, 2023) or generate responses(Chiu et al., 2023)—typically rely on explicit semantic representations.

2.2 TOD systems using Procedural Semantic Representations

One approach to explicit semantics in TOD is procedural semantics (Bollini et al., 2013; Nevens et al., 2024; Verheyen et al., 2023). Procedural semantics offers representations for task descriptions that are specific enough to be executed programmatically and achieve desired results. Ultimately, collaborative agents need executable action representations, but there are potential disadvantages to deriving those representations directly from utterances. Deriving them may involve planning and plan recognition as well as processes of compositional interpretation and resolution of grounded references (Geib et al., 2022). For example, action plans may depend on the capabilities of the agent and the physical state of the environment. An abstract semantic representation can play an important role for collaborative dialogue by representing task content in a way that can be shared across agents and contexts and can mediate between various kinds of linguistic and plan-based reasoning.

[1] https://developer.amazon.com/docs/alexa/custom-skills/send-the-user-a-progressive-response.html

2.3 TOD systems using AMR

AMR is another form of semantic representation used in NLU modules for TOD (Tam et al., 2023). AMR represents each sentence as a rooted, directed, acyclic graph. In the graph, each edge has a label for the relation, and each leaf represents a concept (Banarescu et al., 2013). These graphs can also be written in PENMAN notation (Matthiessen and Bateman, 1992). AMR has been extended to be more suitable for representing dialogues (O'Gorman et al., 2018; Bonial et al., 2020) and multimodal communication (Brutti et al., 2022). Tam et al. (2023) has shown that AMR can be used to annotate actions for both human-human interactions and human-object interactions. AMR has also shown promise in TOD through interactive simulations (Krishnaswamy et al., 2017).

2.4 Minimal Recursion Semantics

We have chosen to use MRS in our work. MRS is a framework that can encode predicate arguments and other grammatical constraints on lexical and phrasal semantics to generate flat semantic representations. An MRS structure is a tuple containing a top handle (GT), a bag of elementary predicates or EPs ("an EP is a single relation with its associated arguments"), and a bag of handle constraints (C) (Copestake et al., 2005). Like AMR, MRS is scalable because it abstracts away from domain-specific content.

While AMR is easy to annotate, and has become a popular semantic representation for text-based tasks, AMR does not support constraint-based ambiguity resolution like MRS does (Copestake et al., 2005; Wein, 2025). The incremental constraint-based approach of MRS also streamlines the representation of dialogue processes such as clarification, thereby facilitating system efforts to ensure common ground. In addition, AMR lacks the full logical expressiveness of MRS (Bender et al., 2015; Bos, 2016), which underpins logical approaches to bridging semantic and common-sense inferences (Hobbs, 1985; Copestake et al., 2005).

We have chosen not to build on existing MRS implementations, such as English Resource Grammar (ERG) (Flickinger et al., 2000) [2], because our approach allows for more flexibility, such as choosing to ignore scopal arguments (which would not have an impact when combining linguistic reasoning and plan-based inferences, since planning modules typ-

[2] https://delph-in.github.io/delphin-viz/demo/

ically do not account for scopal arguments), therefore allowing for a more lightweight and efficient representation. Our MRS implementation is builds on dependency parsing provided by spaCy. This decision is primarily for convenience; dependencies provide a simple and effective starting point for our work. We believe our approach could be adapted as needed to other state-of-the-art real-time dependency or constituency parsers.

3 System Design

For this paper, MRS is used as an early component of NLU to help create a logical form (LF) as a semantic representation that can be used for dialogue systems and updated with information from the planner's inferences, allowing the LF to be updated with information from the environment. Since we are comparing MRS to AMR (to show that if AMR is used in TOD, MRS should be able to do so as well), we will focus on non-scopal EPs, ignoring all EPs that can be a scopal EP (such as adverbs).[3]

3.1 spaCy

For dependency parsing, spaCy was selected due to its popularity and its capability for real-time dependency parsing. It is a transition-based dependency parser that uses an arc-eager system. SpaCy's English models were trained using OntoNotes 5.0 (Weischedel et al., 2013), which contains approximately 1.5 million words from news media, telephone conversations, broadcast conversations, and weblogs. SpaCy's developers report a 95.1% accuracy for unlabeled attachment score (UAS) and 93.7% labeled attachment score (LAS) accuracy when tested on the Penn Treebank (Marcus et al., 1993)[4], which contains articles from the Wall Street Journal (WSJ) from 1984 to 1989. However, a machine learning model evaluated on WSJ may have different accuracy for other domains. We took Cookdial and evaluated predicate-argument relations reported by spaCy and translated to MRS (discussed in Section 4) to determine their consistency with the corresponding Extended-AMR (EAMR). We used spaCy version 3.7.4 with en_core_web_lg model version 3.7.1.

3.2 Implementation

Algorithm 1 shows the logic used to implement MRS to create an LF. It assumes that each word in the sentence is associated with a head, a dependency label, a part-of-speech (POS) tag, and its position in the sentence. Each word may also be associated with a predicate (the meaning carried by the word) and a variable (the discourse referent it evokes).

Algorithm 1 Build MRS LF from Dependencies

Input: *sent* = sentence
Output: *lf*
lf = set()
$ignore\_deps$ ={det,punct,case,adv}
for all $(child, rel, head) \in sent.deps()$ **do**
  if $is\_pred(child)$ **then**
    lf.add $([child.pred, child.var])$
    if $rel \in$ UD_Modifiers **then**
      lf.add$([=, head.var, child.var])$
    if $child.tag =$ VBG **then**
      lf.add$([$nsubj$, child.var, head.var])$
    else if $child.tag =$ VBN **then**
      lf.add$([$dobj$, child.var, head.var])$
  if $rel =$ pobj, dobj **then**
    lf.add$([role(rel, head),$
    $head.head.var, child.var])$
  else if $rel \notin ignore\_deps$ **then**
    lf.add$([rel, head.var, child.var])$
return *lf*

The algorithm loops through each relation in the sentence, focusing on representing the contribution of the dependent element (child). Nouns, pronouns, adjectives, verbs, and auxiliaries without dependents contribute elementary predications. Verbal dependent modifiers assign an appropriate syntactic role to the head referent (subject for present participle, object for past participle). All other modifiers, excluding adverb modifiers which are ignored, equate their variable to the variable of the elementary predicate they are describing. Objects of prepositions are assigned a suitable semantic role with respect to the entity modified by the preposition. Aside from root, determiners, punctuation, adverbs, and case modifiers, all other dependency labels are included in the logical form. Since planning modules typically do not account for scopal arguments, determiners and adverb modifiers have been excluded from consideration.

For the sentence "Pour cranberry juice into a 5-cup ring mold", the MRS algorithm will go through each relation given by spaCy (as shown in Figure 1). If the first dependency identified is the direct object

[3]Note that an entire MRS structure can generally be created with a dependency tree parse.

[4]https://spacy.io/usage/facts-figures

Token	Relation	Part of Speech	Tag	Head	Children	Ancestors
Pour	root	VERB	VB	Pour	[juice,into]	[ ]
cranberry	compound	NOUN	NN	juice	[]	[juice, Pour]
juice	dobj	NOUN	NN	Pour	[cranberry]	[Pour]
into	prep	ADP	IN	Pour	[mold]	[Pour]
a	det	DET	DT	mold	[]	[mold, into, Pour]
5	nummod	NUM	CD	cup	[ ]	[cup, mold, into, Pour]
-	punct	PUNCT	HYPH	cup	[ ]	[cup, mold, into, Pour]
cup	compound	NOUN	NN	mold	[5, -]	[mold, into, Pour]
ring	compound	NOUN	NN	mold	[ ]	[mold, into, Pour]
mold	pobj	NOUN	NN	into	[a, cup, ring]	[into, Pour]
.	punct	PUNCT	.	Pour	[ ]	[Pour]

Table 1: spaCy parse of "Pour cranberry juice into a 5-cup ring mold."

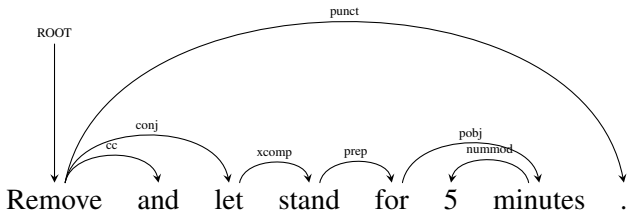


Figure 1: spaCy dependency parse of the sentence: “Remove and let stand for 5 minutes.” Parsed using spaCy.

relationship between the head "Pour" and the child "juice", the algorithm identifies that "juice" evokes a discourse referent, and stores the fact that the predicate "juice" applies to the referent "x_juice_2" by storing [juice, "x_juice_2"] Then, it will identify that the dependency is not a Universal Dependency modifier, and that the dependency is not a preposition, so it will be represented as ["dobj", x_Pour_0, x_juice_2]. This process will be completed while going through all remaining dependencies.

```
(inst-0 / R
  :inform (ac-0-0 "Pour" 3:7/ AC
    :ppt (ing-0 "cranberry juice" 8:23 / FOOD)
    :gol(tool-0-0 "a 5-cup ring mold" 29:46 / TOOL)
    :_result (juice-in-mold))
```

Figure 2: EAMR representation of the instruction “Pour cranberry juice into a 5-cup ring mold.”

For the sentence "Remove and let stand for 5 minutes.", the MRS algorithm will go through each relation given by spaCy (as shown in Figure 4). It will identify and store the elementary predications, "remove", "let", "stand", "5", "minutes" as before. For example, "remove" will be stored as [remove, "x_remove_0"]. It will stores additional relations such as noting the nummod relation between "minutes" and "5" as [’=’, ’x_5_5’, ’x_minutes_6’]. The remaining relations are from the else if clause on line 17. These relations are: [’cc’, ’x_Remove_0’, ’and’], [’conj’, ’x_Remove_0’, ’x_let_2’], [’xcomp’, ’x_let_2’, ’x_stand_3’], [’for’, ’x_stand_3’, ’x_minutes_6’]. When supplied to reference resolution and clarification module, we can potentially recognize that "remove" and "stand" concern an implicit object derived from dialogue context. When combined with a planner module, the planner could infer how to achieve the successive "remove" and "stand" tasks with suitable planner actions.

```
(inst-8 / R
  :inform (ac-8-0 "Remove" 3:9/ AC
    :ppt (NULL / FOOD)
    :ppt (NULL / FOOD)
  :inform (ac-8-0 "stand" 18:23/ AC
    :duration (dur-8-0 "5 minutes"@28:37 / DUR)
    :ppt (NULL / FOOD)))
```

Figure 3: EAMR representation of the instruction “Remove and let stand for 5 minutes.”

4 Evaluation

For our evaluation we chose the Cookdial dataset (Jiang et al., 2022). The data set contains Extended-AMR (EAMR) annotations of recipe instructions, which mimic many ideas and notations from AMR

(Jiang et al., 2022). EAMR uses PENMAN notation (the string and index annotations are placed into ":name" or ":named"), and represents a directed acyclic graph composed of nodes (the entity type) and edges (relation between the predicate and its arguments) (Jiang et al., 2022). For the purposes of evaluation, we will be only considering EAMR with multiple edges or nodes, since EAMR of just a single node would not have any significant information to compare against spaCy's parse, as the entire sentence would be the constituent. This provides us with 227 sentences, totaling 951 constituents to evaluate for the consistency of predicate-argument relations in EAMR captured in both the spaCy parse and MRS clauses.

4.1 Predicate-Argument Consistency

We recursively iterate through the AMR graph, starting from its root node (Algorithm 2 in Appendix), and verify if each constituent has exactly one semantic relation with a different constituent (Algorithm 3 in Appendix). This is done by identifying and counting the external semantic relations the constituent has, and by verifying the alignment of AMR with the dependency head relation (that would be provided to MRS) by spaCy's parse. For example, if you consider Figure 2, the phrase "cranberry juice", we would confirm that there is only one external semantic relation, which in this case would be the head verb "Pour". This means no additional dependencies link to a word in the phrase from elsewhere in the AMR graph, therefore showing that the EAMR and spaCy parse are consistent. This evaluation can be applied across any AMR that contains multiple edges or nodes by following the same methods.

4.2 Evaluation Results

Out of the 951 edges evaluated, it was found that 56 had inconsistent constituency ($\approx 5.9\%$). This performance ($\approx 94.1\%$) is comparable with spaCy RoBERTa (2020) dependency parsing accuracy on Penn Treebank (Marcus et al., 1993), which is 95.1% for unlabeled attachment score.[5] Note that spaCy had incorrectly interpreted "in." as the end of a sentence for two utterances; therefore, it was decided "inch" would be substituted for "in." While this analysis of the consistency of the Dependency Parser's and MRS algorithm highlights specific limitations of the parser, the implications of the dependency parser's accuracy for the LF are not yet fully understood.

[5] https://spacy.io/usage/facts-figures

5 Conclusion

In this paper, we have built on existing AMR annotations to argue that MRS may also be used for semantic representations in TOD. We showed how to evaluate MRS by comparing predicate-argument relations in the input of MRS to those annotated in EAMR for a cooking domain. Evaluation shows that MRS aligns with EAMR relations with 94.1% accuracy when using spaCy's dependency parsing as the main input for our MRS algorithm.

In future work, we plan to explore further uses of MRS as structured, semantic representations to bridge language-based and plan-based inferences for TOD. We hope to develop a versatile NLU module that can be used across multiple domains and even languages—since the Universal Dependencies framework provides consistent cross-linguistic grammar annotations (de Marneffe et al., 2021). We further hope to build on strategies from Traum (1995) and Rich et al. (2001) to allow for tracking and maintaining common ground in collaborative interactions. Finally, we are interested in using our MRS module for coordinating activity by extending our existing implementation of plan filtering and semantic grounding using planning and plan recognition (Geib et al., 2022).

Limitations

While this paper evaluates the dependency parse on EAMR, only relations between EAMR nodes are tracked, leaving out node-internal relations, such as the relation between "cranberry" and "juice" in the EAMR constituent "cranberry juice". Also, while our NLU module may be applicable across domains, it will still require planning modules that may have to be created for each domain, as well as a knowledge base for each domain to identify action types and resolve references. We have also not demonstrated the impact of our techniques on dialogue quality or task success.

Acknowledgments

Thanks to Rich Magnotti and the reviewers for helpful feedback. Supported by NSF awards 2021628, 2119265, and 2427646.

References

Gregor Bachmann and Vaishnavh Nagarajan. 2024. The pitfalls of next-token prediction. *Preprint*, arXiv:2403.06963.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 239–249, London, UK. Association for Computational Linguistics.

Mario Bollini, Stefanie Tellex, Tyler Thompson, Nicholas Roy, and Daniela Rus. 2013. Interpreting and executing recipes with a cooking robot. In *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, pages 481–495. Springer.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Johan Bos. 2016. Squib: Expressive power of Abstract Meaning Representations. *Computational Linguistics*, 42(3):527–535.

Richard Brutti, Lucia Donatelli, Kenneth Lai, and James Pustejovsky. 2022. Abstract Meaning Representation for gesture. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1576–1583, Marseille, France. European Language Resources Association.

Justin Chiu, Wenting Zhao, Derek Chen, Saujas Vaduguru, Alexander Rush, and Daniel Fried. 2023. Symbolic planning and code generation for grounded dialogue. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7426–7436, Singapore. Association for Computational Linguistics.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan Sag. 2005. Minimal recursion semantics: An introduction. *Reseach On Language And Computation*, 3:281–332.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Wenjie Dong, Sirong Chen, and Yan Yang. 2025. ProTOD: Proactive task-oriented dialogue system based on large language model. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9147–9164, Abu Dhabi, UAE. Association for Computational Linguistics.

Yue Feng, Yang Wang, and Hang Li. 2021. A sequence-to-sequence approach to dialogue state tracking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1714–1725, Online. Association for Computational Linguistics.

Dan Flickinger, Ann Copestake, and Ivan A. Sag. 2000. *HPSG Analysis of English*, pages 254–263. Springer Berlin Heidelberg, Berlin, Heidelberg.

Christopher Geib, Denson George, Baber Khalid, Richard Magnotti, and Matthew Stone. 2022. An integrated architecture for common ground in collaboration. *ACS 2022*.

Jerry R. Hobbs. 1985. Ontological promiscuity. In *23rd Annual Meeting of the Association for Computational Linguistics*, pages 60–69, Chicago, Illinois, USA. Association for Computational Linguistics.

Vojtěch Hudeček and Ondrej Dusek. 2023. Are large language models all you need for task-oriented dialogue? In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–228, Prague, Czechia. Association for Computational Linguistics.

Bowen Jiang, Yangxinyu Xie, Zhuoqun Hao, Xiaomeng Wang, Tanwi Mallick, Weijie J Su, Camillo Jose Taylor, and Dan Roth. 2024. A peek into token bias: Large language models are not yet genuine reasoners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4722–4756, Miami, Florida, USA. Association for Computational Linguistics.

Yiwei Jiang, Klim Zaporojets, Johannes Deleu, Thomas Demeester, and Chris Develder. 2022. Cookdial: a dataset for task-oriented dialogs grounded in procedural documents. *Applied Intelligence*, 53(4):4748–4766.

Nikhil Krishnaswamy, Pradyumna Narayana, Isaac Wang, Kyeongmin Rim, Rahul Bangar, Dhruva Patil, Gururaj Mulay, Ross Beveridge, Jaime Ruiz, Bruce Draper, and James Pustejovsky. 2017. Communicating and acting: Understanding gesture in simulation semantics. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS) — Short papers*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Christian M.I.M. Matthiessen and John A. Bateman. 1992. Text generation and systemic-functional linguistics: Experiences from english and japanese.

Ida Momennejad, Hosein Hasanbeig, Felipe Vieira, Hiteshi Sharma, Robert Osazuwa Ness, Nebojsa Jojic, Hamid Palangi, and Jonathan Larson. 2023. Evaluating cognitive maps and planning in large language models with cogeval. *Preprint*, arXiv:2309.15129.

Jens Nevens, Robin De Haes, Rachel Ringe, Mihai Pomarlan, Robert Porzel, Katrien Beuls, and Paul Van Eecke. 2024. A benchmark for recipe understanding in artificial agents. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 22–42.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

OpenAI. 2022. Introducing ChatGPT. `https://openai.com/blog/chatgpt`.

OpenAI. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Charles Rich, Candace L Sidner, and Neal Lesh. 2001. Collagen: Applying collaborative discourse theory to human-computer interaction. *AI magazine*, 22(4):15–15.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2023. Mars: Modeling context state representations with contrastive learning for end-to-end task-oriented dialog. *Preprint*, arXiv:2210.08917.

Christopher Tam, Richard Brutti, Kenneth Lai, and James Pustejovsky. 2023. Annotating situated actions in dialogue. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 45–51, Nancy, France. Association for Computational Linguistics.

David Rood Traum. 1995. *A computational theory of grounding in natural language conversation*. University of Rochester.

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. 2023. Can large language models really improve by self-critiquing their own plans? *Preprint*, arXiv:2310.08118.

Lara Verheyen, Jérôme Botoko Ekila, Jens Nevens, Paul Van Eecke, and Katrien Beuls. 2023. Neuro-symbolic procedural semantics for reasoning-intensive visual dialogue tasks. In *ECAI 2023*, pages 2419–2426. IOS Press.

Shira Wein. 2025. Ambiguity and disagreement in Abstract Meaning Representation. In *Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation*, pages 145–154, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes release 5.0. *Preprint*, Linguistic Data Consortium:2303.08774.

Heng-Da Xu, Xian-Ling Mao, Puhai Yang, Fanshu Sun, and Heyan Huang. 2024. Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2748–2763, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoying Zhang, Baolin Peng, Kun Li, Jingyan Zhou, and Helen Meng. 2023. Sgp-tod: Building task bots effortlessly via schema-guided llm prompting. *Preprint*, arXiv:2305.09067.

Qi Zhu, Christian Geishauser, Hsien chin Lin, Carel van Niekerk, Baolin Peng, Zheng Zhang, Michael Heck, Nurul Lubis, Dazhen Wan, Xiaochen Zhu, Jianfeng Gao, Milica Gašić, and Minlie Huang. 2023. Convlab-3: A flexible dialogue system toolkit based on a unified data format. *Preprint*, arXiv:2211.17148.

A Appendix

A.1 Algorithms

We present Algorithm 2 to show how the AMR graph was traversed while checking relations. We gave each node in a sentence a unique identification, and for each relation in the AMR, we would call Algorithm 3, and report the returned results.

In Algorithm 3, we show how we verify if each constituent has exactly one semantic relation with a different constituent, and how we verify the alignment of the AMR graph and spaCy's parse.

A.2 spaCy Dependency Diagram

Table 1 presents the spaCy dependency parse for the example sentence "Pour cranberry juice into a 5-cup ring mold".

Algorithm 2 AMR_TRAVERSAL

Input: *node_id* = first node id in *amr_graph*, *amr_graph*, *visited* = [], *prev_word=None*
Output: Dataframe updated by *report_result* function
if *node_id* in *visited* **then return**
visited.add(*node_id*)
head_node = *amr_graph*[*node_id*]
for each *child_id* in *head_node.relations* **do**
    child_node = *amr_graph*[*child_id*]
    if *prev_word* $\neq$ *None* **then**
        relations, *head_relations* = *check_relation*(*head_node.words*, *child_node.words*)
        report_result(*relations*, *head_relations*)
    Traverse_AMR(*child_id*, *amr_graph*, *visited*, *node_id*)

Algorithm 3 CHECK_RELATION(WORDS, HEAD_WORDS)

Input: *words*, *head_words*
Output: *relations*, *head_relations*
relations = []
head_relations = []
apart_relations = []
for each *word* in *words*: **do**
    if *word.head* not in *words* **then**
        relations.append(*word.head*)

if len(*relations*) == 1 : **then**
    for *word* in *head_words* : **do**
        if *word* in *relations* **then**
            head_relations.append(*word*)
        else
            ancestors = get_ancestors(*words*, *word*)
            for *ancestor* in *ancestors* **do**
                if *ancestor* == *word and* not in *apart_relation* **and** not in *words* **then**
                    apart_relation.append(*ancestor*)
    if len(*head_relations*) < 1: **then**
        head_relations = *apart_relation*
return *relations*, *head_relations*

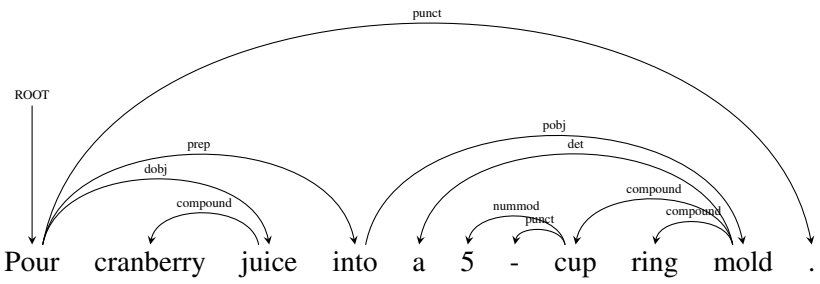


Figure 4: spaCy dependency parse of the sentence: “Pour cranberry juice into a 5-cup ring mold.” Parsed using spaCy.

Evaluation Framework for Layered Meaning Representation

Rémi de Vergnette Maxime Amblard Bruno Guillaume
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{remi.de-vergnette, maxime.amblard, bruno.guillaume}@inria.fr

Abstract

We propose different modular evaluation metrics for Layered Meaning Representation, defined as YARN, a semantic formalism encoded using rich structures that generalize AMR graphs. While existing metrics like SMATCH evaluate graph-based semantic representations such as AMR, they cannot directly handle YARN's more complex structures. We make full use of the modular nature of YARN to propose two families of metrics, depending on the linguistic features and type of semantic phenomenon targeted. The first one, SMATCHY, extends the AMR SMATCH metric. We also propose YARNBLEU, based on the SEMBLEU metric for AMR. We evaluate both families on a small dataset of human annotated YARN structures, adding random modifications simulating annotation mistakes and show that SMATCHY provides a more consistent and reliable approach with respect to the type of modifications considered.

1 Introduction

Evaluating the similarity between two graphs is a non-trivial task, as different approaches emphasize different aspects of structural variation. On the specific topic of graph based semantic formalisms, the most popular metric, SMATCH (Cai and Knight, 2013) compares AMR graphs (Banarescu et al., 2013) by matching nodes from a candidate graph to a reference graph, and treating the task as prediction, evaluating on the popular f-score metric. Alternative metrics based on SMATCH have been proposed like S²MATCH (Opitz et al., 2020) who allows soft matching by incorporating a distance function on concepts. Another popular metric for AMR evaluation is SEMBLEU (Song and Gildea, 2019), which is based on the classical Bleu metric for machine translation, and compares k-grams in the candidate and reference graphs. SMATCH and SEMBLEU have been introduced to take into account the specificities of AMR graphs, and they cannot be applied directly to other kinds of semantic formalism that are not graph-based.

We focus on layered meaning representations such as the recently introduced YARN formalism (Pavlova et al., 2024). YARN is based on AMR, but extends this formalism by adding typed edges and vertices, and enabling certain edges to go from or toward other edges. By allowing one to choose the features they would like to target (like quantification, modalities, aspect), YARN provides a modular framework for partial annotations: it is more expressive than AMR, can represent first-order logic and quantification phenomenon, as well as scope. Meaning representation-based similarity measures have been widely applied to natural language processing tasks, ranging from Natural Language Inference (Opitz et al., 2023) to text generation evaluation (Manning and Schneider, 2021) and compositional semantic similarity measurement (Fodor et al., 2025). Since YARN provides a more complete and accurate representation than AMR, similarity measures on YARN structures have the potential to yield more precise results on such tasks, provided parser accuracy. We propose decomposing the YARN structures as a set of clauses. This allows us to extend the steps presented in the original SMATCH paper to YARN structures. Furthermore, by keeping the information related to edge and vertices types in the clause decomposition, we are able to evaluate the performance of a given parser on various type of phenomenon. We extend SEMBLEU in a similar way, by proposing a way to represent YARN structures as graphs and using the same k-grams extraction method as in SEMBLEU.

We first review the classical AMR metrics SMATCH and SEMBLEU and present YARN. Then, we introduce two metrics families based on SMATCHY and YARNBLEU, and evaluate them on a small dataset of annotated YARN. Finally, we discuss the results and propose future work.

Proceedings of the 6th International Workshop on Designing Meaning Representations, pages 38–48
August 4, 2025.

2 AMR metrics

Smatch (Cai and Knight, 2013) uses a semantically motivated approach, by decomposing the candidate AMR and the reference graph as conjunctions of triples , and computing precision, recall and f-score based on predicting correct triples. Since triples involves variables, the score depends on variable matching of both graphs, and the SMATCH score is calculated as the best f-score over all possible partial one-to-one mapping between the set of variables of the two AMRs. A complete example is given in Appendix A.

SMATCH is an interpretable and semantics-driven metric: each triple represents a predicate in the event structure described by the AMR graph. Thus, it accurately captures the overlap between the two meaning associated to AMRs, in terms of asserted elementary relations between entities or variables. In particular, SMATCH does not heavily penalize incorrect labels: two AMR graphs with similar structure but different vertex labels can still score high if the number of edges outweighs the labels differences. However, using a semantically grounded metric has a cost: finding the optimal variable matching between two AMRs is NP-hard, and SMATCH relies on heuristic, non-deterministic solvers with repeated random initialization.

SemBLEU (Song and Gildea, 2019) on the other hand, does away with variable matching by taking inspiration from the classical BLEU (Papineni et al., 2002) metric and comparing k-grams predicted by the candidate graph to k-grams present in a reference graph. Since BLEU is used to evaluate machine translation, it is motivated by casting AMR parsing as translating from english to AMR. However BLEU cannot be used as is since an AMR graph is not a text sequence. Nevertheless, since BLEU relies on k-grams matching, a straightforward extension of BLEU for graphs has been proposed by (Song and Gildea, 2019) by considering k-grams as sequences of connected k-nodes. More precisely, for a reference graph z and a candidate graph c, SEMBLEU enumerates 1-grams (vertices), 2-grams (labeled edges), ..., n-grams by a traversing both graphs with a breadth-first algorithm, and then applies the standard BLEU equation:

$$\text{BLEU} = e^{min\left(1-\frac{|z|}{|c|},0\right)} \times e^{\sum_{k=1}^{n} w_k \log p_k}$$

$$p_k = \frac{|k\text{-gram}(z) \cap k\text{-gram}(c)|}{|k\text{-gram}(c)|}$$

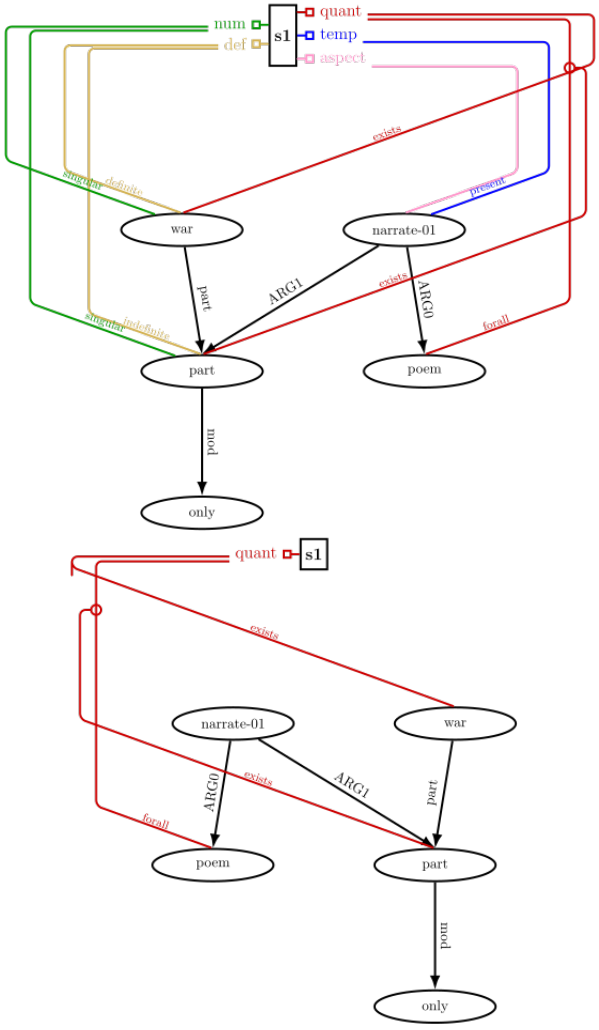


Figure 1: YARN structures representing for "Each poem narrates only a part of the war.". The second structure focuses only on the quantifier feature and the PA part.

Where w is a sequence of n positive parameters summing to 1. The authors of the original SEMBLEU paper use $n = 3$ and $w_1 = w_2 = w_2 = 1/3$.

SEMBLEU has the property of being deterministic and computable in linear time for trees. Although AMR graphs are not necessarily trees, they are generally sparse, and (Song and Gildea, 2019) empirically verified that the property still holds.

3 YARN

In this section, we give a brief overview of the YARN formalism, and how it can be represented as a set of clauses. We refer to (Pavlova et al., 2024) for a more detailed description of the formalism.

The features of YARN that we need to take into account when proposing a metric are the following: The base of a YARN structure is a graph representing the basic predicate argument (PA) structure. (i)

YARN has typed vertices. (ii) YARN has typed edge connecting different types of vertices. (iii) YARN has typed edges[1] connecting vertices or edges to other vertices or edges. (iv) YARN is modular: we might remove all vertices and edges connected to the structure only through feature nodes representing certain features we do not wish to focus on. This allows to get another simplified YARN structure. Figure 1 gives an example of this process.

We use the definition by Pavlova (2025) which, compared to Pavlova et al. (2024), provides a slightly simplified and more expressive version of the YARN formalism. We explicitly define the changes between the former and the later in the following paragraph.

A YARN structure is defined as a 9-tuple:

$$\mathcal{Y} = (S, V, F, D, E, C, L, H, I)$$

Each term denotes a set of labeled edges or vertices. The base of the representation follows AMR: V elements are vertices representing concepts, individuals or attributes, while E edges express relations between V elements. S elements are nodes corresponding to elementary events with F elements, features associated to them. D elements are edges representing discourse relations between elementary events (D is called E_s in Pavlova et al. (2024)). L elements are edges connecting F and V nodes (L is called E_{FV} in Pavlova et al. (2024)). For details on their interpretation and use to model various phenomena, see again Pavlova et al. (2024). The remaining elements are not present in Pavlova et al. (2024): C elements are edges linking V and S nodes to model clauses. H elements are edges going either from elements of F towards other elements of H or L, or from L or other H ones towards V or E. This expresses how features interact and modulate semantic relations between entities. Finally, I are undirected edges between V vertices.

4 SMATCHY

4.1 SMATCHY-BASE

SMATCH uses variables associated to nodes to handle reference towards them, encoding the structure of a graph as a collection of triples. YARN structures can be considered as classical directed graphs that have nodes of different types, with the addition of specific L or H edges that either go from another edge to a node or from a node to an edge. The only missing element in order to use SMATCH on YARN structures would be the ability to encode such edges. This can be done by adding variables corresponding to edges, as illustrated in Figure 2. With this encoding, due to the additional variables assignations, we encode YARN structures as sets of quadruples[2] (corresponding to edges) and triples (corresponding to labels of vertices), or only quadruples by adding dummy variables. An easy extension of SMATCH can then be proposed for YARN, as the best f-score that can be achieved through partial one-to-one variable matching on the clauses (triples and quadruples) defining the given SMATCH structures. We now show how to compute such a matching using integer linear programming (ILP).

ILP formulation let Y_1 and Y_2 be two graph structures, we define V_1 as the set of variables in Y_1, V_2 as the set of variables in Y_2, C_1 the set of clauses appearing in Y_1, C_2 the set of clauses appearing in Y_2.

We say that two clauses are comparable if they correspond to the same type of edge or vertex in the YARN structure, and they are labeled with the same relation, concept or feature type.

We can frame the problem of finding optimal variable alignment as an integer linear programming problem, with the given binary matrixes:

$$v : V_1 \times V_2 \to \{0; 1\} \quad t : C_1 \times C_2 \to \{0; 1\}$$

Where v_{ij} is 1 if and only if variable i is assigned to variable j, and t_{cd} is 1 if and only if the clauses c and d are comparable and match given the variable assignment.

The constraints for v to represent a partial one to one alignment are:

$$\sum_{i=1}^{n} v_{ij} \leq 1, \quad \forall j \in \{1, 2, \ldots, m\}$$

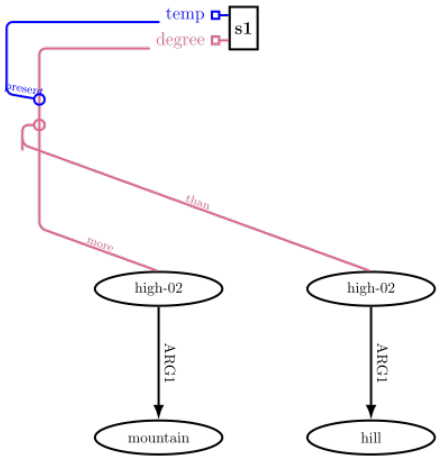
$$\sum_{j=1}^{m} v_{ij} \leq 1, \quad \forall i \in \{1, 2, \ldots, n\}$$

Additionally clauses $c_i \in C_1$ and $c_j \in C_2$ match if they are comparable and their variables match, we can formalize this in the following way: if c_i and c_j are comparable and have respective variables (x, y, z) and (a, b, c) we write:

$$t_{c_i c_j} \leq v_{xa} \quad t_{c_i c_j} \leq v_{yb} \quad t_{c_i c_j} \leq v_{zc}$$

[1] This is a slight abuse of terminology.

[2] We follow SMATCH formulation: a triple correspond to a relation together with two variables, and a quadruple consists of a relation together with three variables.



$\texttt{instance\_f}(\texttt{degree}, d)$
$\texttt{instance\_f}(\texttt{temp}, t)$
$\texttt{instance\_s}(\texttt{event}, s_1)$
$\texttt{instance\_v}(\texttt{high-02}, h)$
$\texttt{instance\_v}(\texttt{high-02}, h_2)$
$\texttt{instance\_v}(\texttt{hill}, h_3)$
$\texttt{instance\_v}(\texttt{mountain}, m)$
$\texttt{feature\_f}(d, s_1)$
$\texttt{feature\_f}(t, s_1)$

$e_1 := \texttt{ARG1\_e}(h, m)$
$e_2 := \texttt{ARG1\_e}(h_2, h_3)$
$l_1 := \texttt{more\_l}(d, h)$
$h_1 := \texttt{present\_h}(t, l_1)$
$h_2 := \texttt{than\_h}(l_1, h_2)$

Figure 2: Expression of a YARN structure representing "Mountains are higher than hills" as triples and quadruples.

$$t_{c_i c_j} \leq v_{xa} \quad t_{c_i c_j} \leq v_{yb}$$

Up to this point we follow closely the formulation of (Cai and Knight, 2013), accounting for additional variables. Most of the edges in a YARN graph are directed or between nodes of different types. The only exception to this rule in YARN structures are I edges that link V vertices and that are undirected. If c_i and c_j correspond to such vertices, linking nodes corresponding to variables x, y and a, b respectively then we may write:

$$t_{c_i c_j} \leq v_{xa} + v_{xb} \quad t_{c_i c_j} \leq v_{ya} + v_{yb}$$

Where the constraints on v insure that both right hand side are less than or equal to 1, and that if they are both 1, then $\{x, y\} = \{a, b\}$.

When clauses c_i and c_j correspond to relations that may not be compared, we write

$$t_{c_i c_j} = 0$$

Naming the set of pairs of matrixes that follow those constraints Λ, finding the best alignment is equivalent to solving the ILP problem:

$$\max_{(t,v)\in\Lambda} \sum_{c_i \in C_1, c_j \in C_2} t_{c_i c_j}$$

Since Λ is not empty (setting v and t equal to 0 satisfies all the constraints) and the function to maximize is bounded by the number of comparable clauses, the problem is well defined and can be solved in reasonable time[3] by ILP solvers.

[3]To give a rough estimate, computing the optimal alignement for a given pair of YARN structures takes about 20 ms on a personal laptop using the CBC solver (Forrest et al., 2024) through the python PuLP(Mitchell et al.) ILP modeling library.

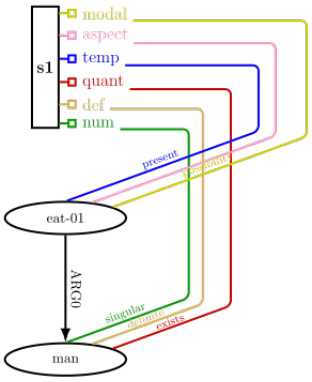


Figure 3: A simple YARN structure achieving high average base SMATCHY score (0.55 average f-score) against human annotated samples for unrelated sentences

Once an optimal mapping is found, we consider clauses that match between the candidate graph and the reference graph as true positives (TP), clauses that are present in the candidate graph and not in the reference graph as false positives (FP), and clauses that are not present in the candidate graph but are present in the reference graph as false negatives (FN): we then compute recall, precision and f-score using the usual formulas (Davis and Goadrich, 2006). Continuing with (Cai and Knight, 2013), we use the f1-score as the final metric.

4.2 Feature Aware SMATCHY

Using the previously introduced metric to compare YARN structures is unsatisfactory. It leads to considering every element of the YARN structure as equally important, either during alignment or phase. For instance, instance clauses predicting the very presence of a feature count as much as clauses relating to how this feature acts on other elements

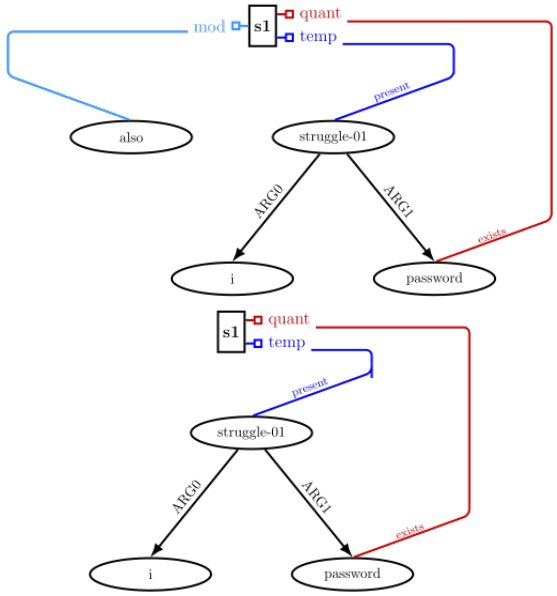


Figure 4: YARN structures for "I also struggle with passwords", before and after filtering *quant* and *temp*

of the structure, which is not ideal. Using an annotated subset of 100 sentences from the English PUD dataset (Zeman et al., 2017), the average score of random pairs of graph was 0.45, which is unsatisfactory, as one would expect this number to be closer to zero. In fact, comparing every annotated graph with a nearly empty YARN graph composed of only two V nodes and several common features gives an average score of 0.55 (see Figure 3).

Additionally, YARN has the advantage of allowing easily one to "switch" features (see Figure 1), depending on what kind of semantic phenomenon they would like to focus on. This should be reflected in any metric evaluating similarity of YARN structures: we would like to have not only a score reflecting how well two structures globally match, but also a family of derived metrics reflecting how they match on certain restricted set of features.

To tackle both challenges, we propose to retain the alignment method of the SMATCHY-BASE metric, but modify the scoring function, in order to ignore certain easy or irrelevant matches. Concretely, once an optimal variable matching is found between the variables corresponding to the two structures, we filter out the set of clauses considered for the precision and recall calculation.

Clause filtering algorithm Given a set of types $\mathbb{T} \subset \{S, V, F, D, E, C, L, H, I\}$, and a set of feature labels $\mathbb{F}$, we filter clauses by: (1) removing instance clauses defining features not in $\mathbb{F}$; (2) recursively removing clauses referencing variables from removed clauses; (3) recursively removing clauses whose variables appear only in removed clauses; and (4) removing clauses of types not in $\mathbb{T}$. Steps 1-3 "switch off" layers, see Figure 1 and Figure 4, while Step 4 filters the clauses considered according to type in order to ignore easy matches.

We now give an example of this filtering process. Let $\mathbb{T} = \{V, E, H, L\}$ and $\mathbb{F} = \{quant, temp\}$. We might choose this setting to evaluate how well a parser can extract first-order logical formulas as well as temporal features.

The YARN structure shown at the top of Figure 4 is split into the following clauses:

(1) $e_1 := \texttt{ARG0\_e}(s, i)$
(2) $e_2 := \texttt{ARG1\_e}(s, p)$
(3) $\texttt{feature\_f}(s_1, m)$
(4) $\texttt{feature\_f}(s_1, q)$
(5) $\texttt{feature\_f}(s_1, t)$
(6) $\texttt{instance\_f}(\texttt{mod}, m)$
(7) $\texttt{instance\_f}(\texttt{quant}, q)$
(8) $\texttt{instance\_f}(\texttt{temp}, t)$
(9) $\texttt{instance\_s}(\texttt{event}, s_1)$
(10) $\texttt{instance\_v}(\texttt{also}, a)$
(11) $\texttt{instance\_v}(\texttt{i}, i)$
(12) $\texttt{instance\_v}(\texttt{password}, p)$
(13) $\texttt{instance\_v}(\texttt{struggle-01}, s_1)$
(14) $l_1 := \texttt{edge\_l}(m, a)$
(15) $l_2 := \texttt{exists\_l}(q, p)$
(16) $l_3 := \texttt{present\_l}(t, s_1)$

Let's apply the four steps of the filtering process.

Step 1 Remove the instance clauses that define feature variables corresponding to features that are not in $\mathbb{F}$: Remove clause (6).

Step 2 Recursively remove the clauses referencing variables defined in clauses that have been removed: Remove clause (3), remove clause (14).

Step 3 Recursively remove clauses whose variables are referenced only in clauses that have been removed: Remove clause (10): thus the set of clause that match the second structure in Figure 4.

Step 4 Remove clauses of types that are not in $\mathbb{T}$: Remove clause (4), (5), (7), (8) and (9).

The final set of clauses is:

(1) $e_1 := \texttt{ARG0\_e}(s, i)$
(2) $e_2 := \texttt{ARG1\_e}(s, p)$
(11) $\texttt{instance\_v}(\texttt{i}, i)$
(12) $\texttt{instance\_v}(\texttt{password}, p)$
(13) $\texttt{instance\_v}(\texttt{struggle-01}, s_1)$

(15) $l_2 := \texttt{exists\_l}(q, p)$

(16) $l_3 := \texttt{present\_l}(t, s_1)$

To be able to compare the general proximity of two YARN structures, we propose using our metric with $\mathbb{T} = \{S, V, D, E, C, L, H, I\}$, that is, removing only clauses of type F, with no filtering on features. With this setting, the average proximity score of pairs of structures taken randomly from our dataset drops to 0.20, while the average score between YARN structures and the structure presented in Figure 3 drops to 0.23. This is on par with results obtained using SMATCH on AMR graphs (Cai and Knight, 2013). We call the metric obtained in this setting SMATCHY-GENERAL. To have a metric focused on the PA substructure of YARN structures, we propose setting $\mathbb{T}$ to $\{V, E\}$. This is very similar to SMATCH, only using the additional more complex YARN elements to guide the variable alignment phase. We call this metric SMATCHY-PA. To evaluate on the fragment of YARN corresponding to first-order logic, we define SMATCHY-FOL by setting $\mathbb{T}$ to $\{S, V, E, H, L\}$ and $\mathbb{F}$ to $\{quant, neg\}$. We may also set $\mathbb{T}$ to $\{S, D\}$ in order to evaluate discourse relations parsing, or to $\{V\}$ for concept and entity recognition.

5 YarnBLEU

We also extend the definition of SEMBLEU to YARN structures. We leverage a graph translation of YARN structures, as seen in Figure 5: every element x of the structure is converted to a typed node $n(x)$, with type in $\{S, V, F, D, E, C, L, H, I\}$. Additionally, for every edge e in the YARN structure connecting two elements x_1 and x_2, we create two unlabeled edges $(n(x_1), n(e))$ and $(n(e), n(x_2))$. Like we did previously with SMATCHY, we propose a family of metrics, depending on the nodes considered. For a set of types $\mathbb{T}$ and features $\mathbb{F}$ applying the same process as in Figure 4.2, we extract a YARN substructure based on $\mathbb{F}$ (step 1 to 3) then select only nodes corresponding to the types in $\mathbb{T}$ before k-grams extraction. We then apply the same formula as SEMBLEU. We build in this fashion the YARNBLEU-GENERAL and YARNBLEU-PA metrics, as well as the YARNBLEU-FOL metrics that are analogous to SMATCHY-GENERAL, SMATCHY-PA and SMATCHY-FOL respectively. Since SEMBLEU additionally depends on n (the maximal size of k-grams considered) and w, we also need to set those parameters. The value proposed by (Song and Gildea, 2019) is w to $(1/3, 1/3, 1/3)$ and n to 3. In order to handle the same range of global dependencies on the PA structure while accounting for additional nodes coming from edges, we set n to 5 and w to $(1/5, ..., 1/5)$.

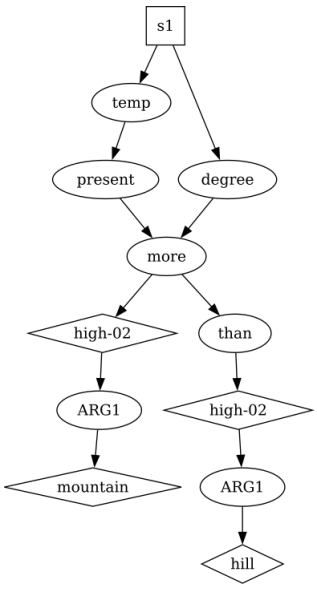


Figure 5: The graph of the YARN structure in Figure 2.

6 Experiments

6.1 Elementary modifications

We first propose a simple evaluation scheme in order to evaluate the general properties of SMATCHY- and YARNBLEU- type metrics with respect to random modifications that simulate annotator errors. We do not cover every type of mistake and the wide range of possible annotation errors. Our main focus is sensitivity and bias, as we want to measure how errors of different forms are differently penalized by such metrics. In particular, SMATCHY-GENERAL should not penalize overly one type of errors, while SMATCHY-PA should mostly penalize errors in the PA substructure of a YARN structure.

We evaluate our more fine-grained first-order oriented metrics SMATCHY-FOL and YARNBLEU-FOL on the same dataset. When changing the labels of V or E elements, we carefully introduce a new distinction. As can be seen with the node labeled "also" in Figure 4, some elements of the YARN structure will not be present in the first-order formula that can be extracted from a given YARN structure: this is typically the case for modifiers acting on elementary events. We tag such elements as "first order irrelevant" (FOI). Other elements are tagged "first order relevant" (FOR). As the con-

version of YARN structures to logical formulas is not the focus of this paper, we do not elaborate on the specific procedure one would use to build such formulas. We then compare the scores obtained when changing the label of FOR elements or FOI elements. Furthermore, modification of L and H edges are also separated between those that act on edges spanning from quantification and negation features (considered FOR) and the others (considered FOI), as only the former will have an influence on the final logical formula. The results are shown in Figure 6. As we can see, SMATCHY-FOL and YARNBLEU-FOL are able to distinguish between those two types of modifications, and only penalize acting on FOR elements. However, the general trend of fuzzier and more biased distributions for YARNBLEU metrics is still present, with YARNBLEU-FOL penalizing more modification of E edges than H or L edges.

6.2 Random chain of modifications

As a way to simulate the influence of more substantial annotation errors, we now apply sequences of random transformations to the YARN structures. This setup complements the first analysis by evaluating how metric scores degrade across cumulative and structured perturbations, rather than isolated changes. Our transformations consist in changing labels, and adding or removing random elements of types E, V, F, L, H. Those transformations are not elementary as we keep valid YARN structures at each transformation step: if a feature F is removed, we remove elements that are attached to the main structure only through this feature.[4] In the same spirit, adding a new V element, also adds an E edge linking it to the main structure. We thus keep track of the number of elementary modification (insertion, deletion of an element or change of a label) performed. We check that restricting the type of modifications to FOI elements doesn't imply a drop in YARNBLEU-FOL and SMATCHY-FOL.[5] We compute the score of the modified structures with respect to the original ones, and plot the scores as a function of the number of elementary modifications performed for SMATCHY-GENERAL, SMATCHY-PA, YARNBLEU-GENERAL and YARNBLEU-PA. The results for a small number of trajectories are shown in Figure 7. SMATCHY metrics degradation follow the editing distance more regularly than YARNBLEU. In particular, we observe mostly non increasing trajectories for SMATCHY, while this is not the case for YARNBLEU.

We note that the occasional increases observed in YARNBLEU scores is still present when changing the value of the n and w parameters. This seems to come from the precision oriented approach of SEMBLEU and YARNBLEU: removing valid elements from a modified structure might increase scores if those elements are linked to wrong ones, as it might reduce drastically the amount of wrong predicted k-grams. It is the role of the brevity penalty factor to counter this kind of effects, but it is not always sufficient: the formula proposed by (Song and Gildea, 2019) seems to rely on the assumption that AMR graphs are sparse enough that the number of k-grams extracted from them grows linearly with size of the graph: while this has been heuristically verified by the same authors on existing AMR datasets, it is not the case for YARN structures.

7 Discussion

The observed behavior of SMATCHY and YARNBLEU in our evaluation protocol leads us to favor SMATCHY for its more predictable and controlled response to parsing or annotation errors. SEMBLEU is a biased measure that penalizes mistakes differently across various regions of a graph, depending on local connectivity patterns. This bias is even more pronounced for YARN than for AMR, as complex YARN structures exhibit very different topological properties in the (H, L) substructure compared to the rest of the structure, due to specific constraints on these elements. Additionally, as noted earlier, the brevity penalty proves insufficient to address these issues.

Are there still reasons to favor SEMBLEU family metrics like YARNBLEU? The main argument appears to be computational complexity, as YARNBLEU can be computed without requiring a variable alignment phase. However, alternative solutions exist that arguably provide better approaches to assessing graph similarity (Kachwala et al., 2024; Sun and Xue, 2024; Shou and Lin, 2023). By focusing on elementary modifications, we evaluate semantic similarity on architectural grounds. YARNBLEU exhibits bias toward penalizing errors more heavily in highly connected regions of the graph, which may occasionally be desirable: in the same way AMR top elements correspond to main verbs and their core arguments, highly connected regions

[4] As a consequence, removing the quantification feature will also remove every H or L edge expressing quantification.

[5] Not obvious as FOI elements still influence the alignment.

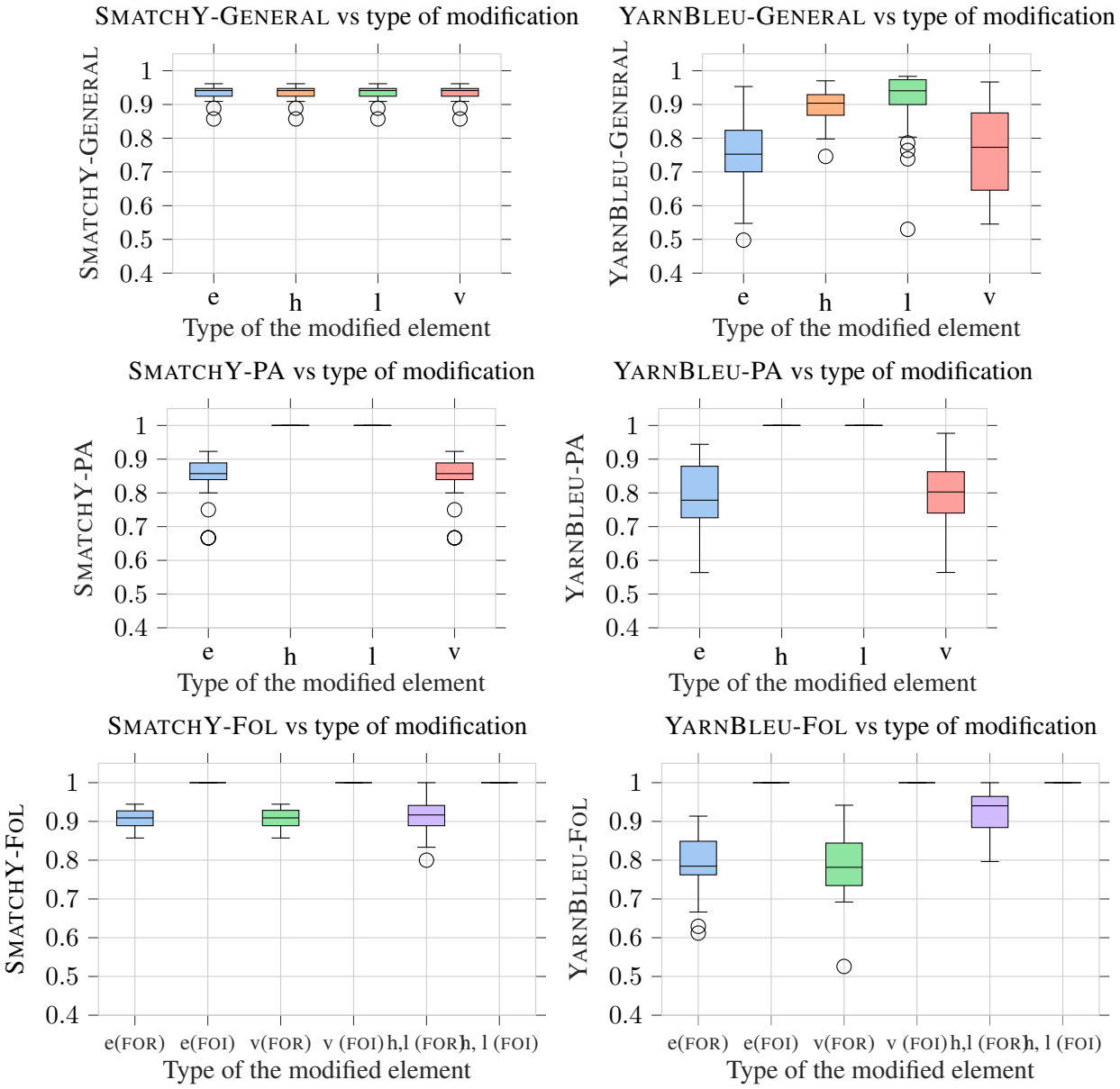


Figure 6: Distribution of scores for the metrics SMATCHY-GENERAL, YARNBLEU-GENERAL, SMATCHY-PA, YARNBLEU-PA, SMATCHY-FOL and YARNBLEU-FOL

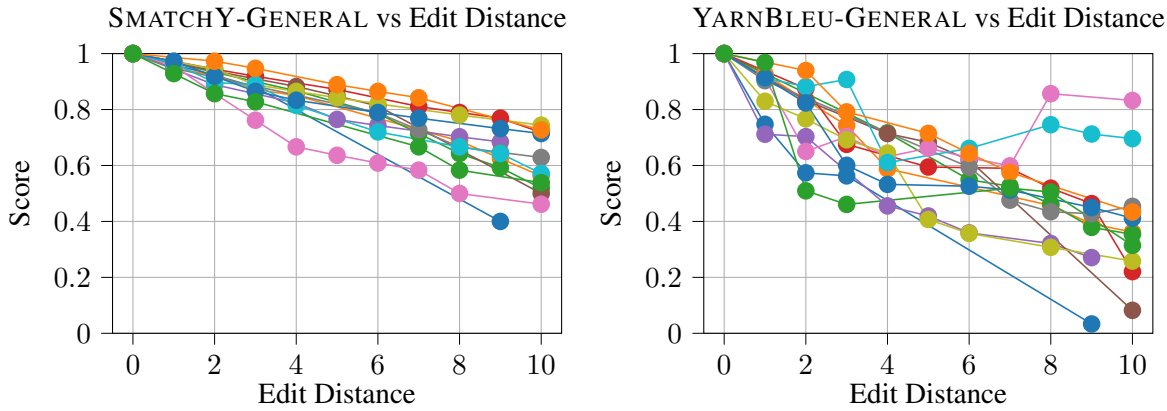


Figure 7: SMATCHY-GENERAL and YARNBLEU-GENERAL scores as functions of the number of modifications performed on the original YARN structure. Colors correspond to different sequences of modifications or original structures.

in YARN structures might correspond to elements that are more important for sentence interpretation.

To apply SEMBLEU to YARN structures, we leverage a graph translation approach. It would also be possible to apply SMATCH directly to YARN structures using this graph translation; however, we argue this is undesirable as it would represent a quadruple $a := \text{rel}(b, c)$ as three triples: instance(a, rel), $\phi(b, a)$, and $\phi(a, c)$. While this formulation suffices for checking isomorphy, it is problematic for fine-grained similarity evaluation. By tripling the number of clauses related to edges, it breaks the symmetry between nodes and edges. Additionally, compared to the quadruple formulation, this approach allows partial matching of the original edge (matching only source or target), which is unsatisfactory.

8 Conclusion

Providing evaluation metrics is a necessary first step toward the development of semantic parsers. In this context, we have introduced a new family of metrics tailored to the evaluation of parsing over YARN structures, derived from SMATCH and SEMBLEU. We have shown how to extend those original metrics to handle the specificities of YARN structures, and how to use it to evaluate parsing on different structural aspects. Those include the core predicative kernel of the structure with SMATCHY-PA and YARNBLEU-PA, the general relatedness with SMATCHY-GENERAL and YARNBLEU-GENERAL, or the first-order logic aspect with SMATCHY-FOL and YARNBLEU-FOL. We have shown that our metrics are able to distinguish and penalize different types of modifications on YARN structures. Our results suggest that using alignment based methods similar to SMATCH provide a more robust way of evaluating parsing on formalisms such YARN structures, as they seem to be less biased and more predictable than graph traversal methods such as SEMBLEU. We emphasize on the fact that many other metrics can be derived from the SMATCHY and YARNBLEU framework, allowing to focus on very specific aspects of semantic parsing, and to evaluate the overall performance or abilities of different type of models on those aspects. This results from the structural richness of YARN structures, which can be used to model a broad variety of phenomena. Furthermore, the extreme modularity of YARN allows for many applications: A single YARN annotated dataset is enough to evaluate capacities of parsers and language models across many tasks, from named entity recognition and word sense disambiguation to parsing of AMR like structures, first-order logic formulas, discourse relations and more simply by switching the evaluation metrics.

9 Limitations

The metrics we present inherit the same limitations as the ones they are based on. We can hypothesise that SMATCHY scoring systems neglect small but semantically relevant structural differences, leading to high scores for unacceptable parses, as was observed with SMATCH in (Opitz and Frank, 2022). A direction for future research is to align with human judgment by learning to aggregate different SMATCHY or YARNBLEU scores, using various choices of $\mathbb{F}$ and $\mathbb{T}$, with optimized weighting coefficients. In addition, the absence of soft concept matching penalizes structures that contain closely related but not identical concepts, overlooking nuanced semantic similarities. This limitation has been criticized and addressed in previous work on SMATCH and SEMBLEU (Opitz et al. (2020), Opitz et al. (2021)). Future work could explore incorporating soft matching in order to provide more permissive metrics evaluating semantic relatedness of YARN structures.

Furthermore, the evaluation protocol presented in this paper is biased in favor of SMATCHY because it focuses on a restricted set of modifications that induce a high variability on high level structural features of the structure as captured by YARNBLEU k-grams while leaving the underlying SMATCHY variable alignment largely unaffected.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine*

Learning, ICML '06, page 233–240, New York, NY, USA. Association for Computing Machinery.

Jennifer Fodor, Simon De Deyne, and Shohei Suzuki. 2025. Compositionality and sentence meaning: Comparing semantic parsing and transformers on a challenging sentence similarity dataset. *Computational Linguistics*.

John Forrest, Ted Ralphs, Stefan Vigerske, Haroldo Gambini Santos, John Forrest, Lou Hafer, Bjarni Kristjansson, jpfasano, EdwinStraver, Jan-Willem, Miles Lubin, rlougee, a andre, jpgoncal1, Samuel Brito, h-i gassmann, Cristina, Matthew Saltzman, tosttost, Bruno Pitrus, Fumiaki MATSUSHIMA, Patrick Vossler, Ron @ SWGY, and to st. 2024. coin-or/cbc: Release releases/2.10.12.

Zohair Kachwala, Jisun An, Haewoon Kwak, and Filippo Menczer. 2024. REMATCH: Robust and Efficient Matching of Local Knowledge Graphs to Improve Structural and Semantic Similarity. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Emma Manning and Nathan Schneider. 2021. Referenceless parsing-based evaluation of AMR-to-English generation. In *Eval4NLP Workshop*.

Stuart Mitchell, Anita Kean, Andrew Mason, Michael O'Sullivan, Antony Phillips, and Franco Peschiera and. Optimization with pulp.

Juri Opitz, Angel Daza, and Anette Frank. 2021. Weisfeiler-leman in the bamboo: Novel amr graph metrics and a benchmark for amr graph similarity. *Transactions of the Association for Computational Linguistics*, 9:1425–1441.

Juri Opitz and Anette Frank. 2022. Better Smatch = better parser? AMR evaluation is not so simple anymore. In *Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems*, pages 32–43, Online. Association for Computational Linguistics.

Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. AMR Similarity Metrics from Principles. *Transactions of the Association for Computational Linguistics*, 8:522–538.

Juri Opitz, Sebastian Wein, Julia Steen, Anette Frank, and Nathan Schneider. 2023. AMR4NLI: Interpretable and robust NLI measures from semantic graphs. In *Proceedings of the International Conference on Computational Semantics (IWCS)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Siyana Pavlova. 2025. *Tools and methods for semantically annotated corpora*. Ph.D. thesis, Université de Lorraine.

Siyana Pavlova, Maxime Amblard, and Bruno Guillaume. 2024. YARN is All You Knit: Encoding Multiple Semantic Phenomena with Layers. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 66–76, Torino, Italia. ELRA and ICCL.

Zeyu Shou and Fangzhao Lin. 2023. Evaluate AMR graph similarity via self-supervised learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

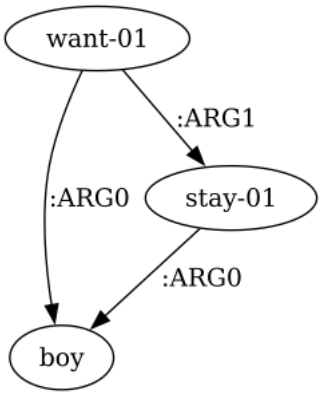
Linfeng Song and Daniel Gildea. 2019. SemBleu: A Robust Metric for AMR Parsing Evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4547–4552, Florence, Italy. Association for Computational Linguistics.

Haibo Sun and Nianwen Xue. 2024. Anchor and broadcast: An efficient concept alignment approach for evaluation of semantic graphs. In *Proceedings of the International Conference on Language Resources and Evaluation and the Conference on Computational Linguistics (LREC-COLING)*.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

A Appendix

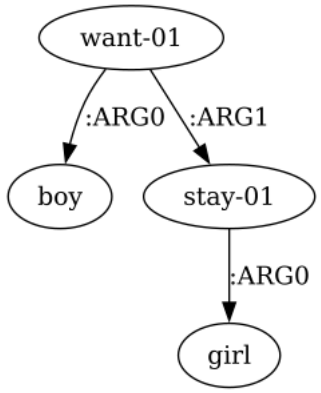
AMR graph for "the boy wants to stay"



and its triplet decomposition:

$$\texttt{instance}(w, \texttt{want-01}) \wedge$$
$$\texttt{instance}(b, \texttt{boy}) \wedge$$
$$\texttt{instance}(s, \texttt{stay-01}) \wedge$$
$$\texttt{ARG0}(w, b) \wedge$$
$$\texttt{ARG0}(s, b) \wedge$$
$$\texttt{ARG1}(w, s)$$

The same for the sentence: "the boy wants the girl to stay"



$$\texttt{instance}(x, \texttt{want-01}) \wedge$$
$$\texttt{instance}(y, \texttt{boy}) \wedge$$
$$\texttt{instance}(z, \texttt{stay-01}) \wedge$$
$$\texttt{instance}(a, \texttt{girl}) \wedge$$
$$\texttt{ARG0}(x, y) \wedge$$
$$\texttt{ARG0}(z, a) \wedge$$
$$\texttt{ARG1}(x, z)$$

Highlighted triples reflect variable alignment: blue for matching, red for non-matching. SMATCH score between the two AMR is 0.77.

Variable (First AMR)	**Matching Variable (Second AMR)**
w	y
b	y
s	z
(None)	a

Representing ISO-Annotated Dynamic Information in UMR

Kiyong Lee
Korea University, Seoul
ikiyong@gmail.com

Harry Bunt
Tilburg University, NL
harry.bunt@tilburguniversity.edu

James Pustejovsky
Brandeis University, Waltham
jamesp@brandeis.edu

Alex C. Fang
City University of Hong Kong
alex.fang@cityu.edu.hk

Chongwon Park
University of Minnesota Duluth
cpark2@d.umn.edu

Abstract

The ISO working group on semantic annotation aims to adopt the UMR formalism to represent dynamic information involving motions and their embedding grounds. The paper details how ISO's XML-based temporal and spatial annotations, involving motions and spatio-temporally conditioned event-paths, will be converted to AMR or UMR forms. It also attempts to enrich the representation of dynamic information with the integrated spatio-temporal annotation scheme that accommodates first-order dynamic logic, as briefly noted. The main motivation of such an effort is to make spatio-temporal annotations and related dynamic information easily understandable by artificial agents like robots to act. Our approach bridges ISO's richly specified standards with the task-oriented expressiveness of UMR and dynamic logic. This integration paves the way for seamless downstream use of spatio-temporal annotations in dialogue systems, simulation environments, and embodied agents.

Key Words: dynamic information, dynamic space, embedding ground, motion, spatio-temporal annotation, UMR

1 Introduction

We propose and explore the use of UMR in ISO's new project on *motion in dynamic space* (ISO/PWI 24617-18). Given Pustejovsky et al. (2019)'s use of AMR, the adoption of AMR for ISO's annotation standards is not novel. Furthermore, the adoption of AMR or UMR has been motivated by the rapid rise in their use in computational linguistics over the past decade; they simplify computational annotation processes while maintaining scalability, unencumbered by extensive syntactic pre-analysis.

As pointed out in Pustejovsky et al. (2019), the strength of AMR lies in its focus on the *predicative* core of a sentence while presenting an intuitive representation for semantic interpretation. More importantly, treating predicates as the root of each AMR structure facilitates annotation processes, just as the event-based temporal annotation of ISO-TimeML and the motion-based spatial annotation of ISO-Space are anchored to eventuality and motions, respectively.

The proposed project's scope for annotating motions embedded in spatio-temporal domains encompasses motions, space, time, and the embedding ground of a motion, called *dynamic space*. We aim to enrich this annotation scheme by augmenting the categorization of spatial and temporal entities with first-order dynamic logic and an iterative program procedure.

The paper will develop as follows. We discuss representing semantic annotations of language in Section 2. In Section 3, we demonstrate how ISO's dual annotation structures are represented in UMR. Section 4 introduces Spatio-Temporal Markup Language (Pustejovsky and Moszkowicz, 2011) and Generative Lexicon-based AMR (GLAMAR) (Tu et al., 2024) to treat motion-oriented dynamic information with the notion of sub-events. The dynamic logic formulates constraints on the iterative process of motions. The paper ends with concluding remarks.

2 Representing Semantic Annotations of Language

2.1 Abstract Annotation Scheme vs Concrete Physical Representation Format

Following Bunt (2010), the ISO SemAF group has divided the specification of each annotation scheme into two sub-components. The first sub-component *abstract syntax* formally defines the annotation structures of the scheme in abstract (set-theoretic) terms while reflecting its conceptual design based on a metamodel. In contrast, the other sub-component, *concrete syntax*, has adopted XML as the physical format for representing annotation

Proceedings of the 6th International Workshop on Designing Meaning Representations, pages 49–58

structures. As depicted in Figure 1, a variety of concrete syntaxes is possible for representing annotation structures. Still, each of them must conform to the proposed abstract syntax while ideally retaining their logical equivalence. Hence, each concrete specification of representing annotation structures depends totally on the abstract syntax of an annotation language.

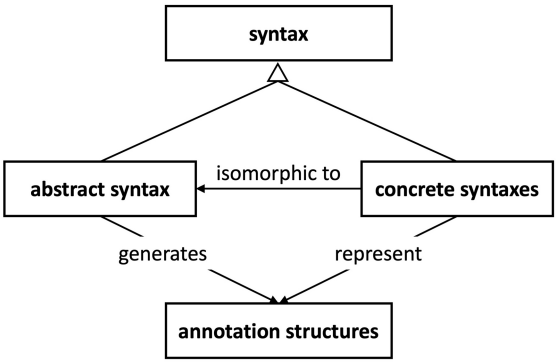


Figure 1: Syntax of an Annotation Language: Abstract vs. Concrete (Lee, 2023)

While introducing the two ISO standards, ISO-TimeML and ISO-Space, Pustejovsky (2017a) and Pustejovsky (2017b) have adopted two different representation formats. XML was adopted to represent annotation structures in ISO-TimeML, but a predicate-logic-like format was adopted in ISO-Space. Nevertheless, the representation of annotation structures in both representation formats conforms to their respective abstract specifications (syntaxes) of temporal and spatial annotations.

Example (refexTS briefly shows how they represent annotation structures.

(1) a. Data with categorized identifiers:
$\texttt{John}_{se1}$ $\texttt{left}_{e1/m1}$ $\texttt{Boston}_{pl1}$
$\texttt{yesterday}_{t1}$.

b. ISO-TimeML (Pustejovsky, 2017a):
```
<EVENT id="e1" target="w2"
pred="LEAVE" tense="PAST"/>
<TIMEX3 id="t1" target="w4"
type="DATE" value="2025-02-16"/>
<TLINK eventID="e1"
relatedToTime="t1"
relType="IS_INCLUDED"/>
```

c. ISO-Space (Pustejovsky, 2017b):
```
SPATIAL_ENTITY(id=se1,
type=PERSON,form=NAM)
MOTION(id=m1, target=w2,
motion_class=LEAVE, tense=PAST)
PLACE(id=pl1, target=w3,
cvt=CITY, form=NAM)
MOVELINK(id=mvli, trigger=m1,
mover=se1, source=pl1)
```

Both ISO-TimeML and ISO-Space focus on predicates, which can be either events or motions. `TLINK` relates the event of leaving to the time *yesterday*. Triggered by the motion $left_{m1}$, `MOVELINK` relates the spatial entity $John_{se1}$ to the source $Boston_{pl1}$.

2.2 UMR as a New Representation Format

UMR adopts the AMR formalism but extends its sentence-level representation to the document level (UMR, 2022). Consider first the sentence-level representation as in Example 2.

(2) a. Data:
```
(s / sentence
(The man left Boston yesterday
before it rained.))
```

b. AMR Format:
```
(l / leave-01
  :ARG0 (m / man)
  :source (b / Boston)
  :temporal (y / yesterday)
  :temporal (b1 / before
    :op1 (r / rain))
```

The AMR formalism represents abstract semantic concepts and relations that include event participant roles, such as `ARG0` or `actor`. In the AMR format, as in (2b) above, the slash (`/`) indicates semantic *concepts* while the colon (`:`) indicates a value of a semantic *relation*. In addition to argument roles, these relations form triplets bound to a governing concept (e.g., `l / leave-01 :ARG0 (m / man)`).

UMR then adds a document-level representation to the sentence-level representation. For example, the sentence-level representation can be extended to a document-level representation such as Example 3 be added:

(3) UMR Document-level Representation
```
(s / sentence)
(d / document-level
   :temporal (sr :before sl))
```

Linked to the sentence-level representation (2), the document-level representation (3) relates the rain

event sr to the event of John's departure sl, interpreted as stating that John's departure occurred before the rain.

3 Representing ISO's Dual Annotation Structures in UMR

3.1 Dual Structures of Annotation

ISO's SemAF annotation schemes formally define annotation structures, each divided into two substructures: *entity structures* and *link structures*. Entity structures are anchored to markables in segmented communicative or textual data while marking them up for specific purposes, such as annotating temporal or spatial information in language. In contrast, link structures each relate an entity structure to a set of other entity structures.

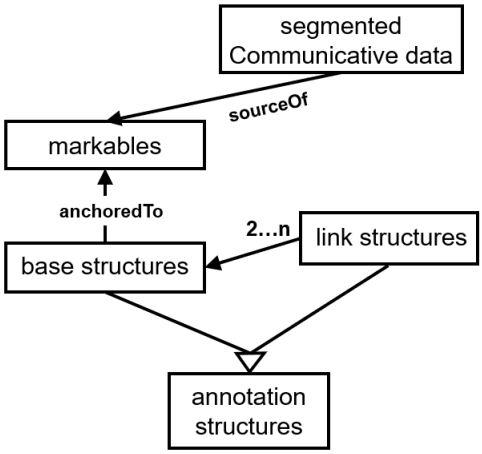


Figure 2: Two-level Annotation Structures

3.2 Temporal Link

In ISO-TimeML, the temporal link relates two entity structures annotating events temporally.

(4) a. John left$_{e1}$ before$_{s1}$ it rained$_{e2}$.

b. Temporal Annotation:

```
<EVENT id="e1", target="w2"
pred="LEAVE"/>
<SIGNAL id=""s1', target="w3"/>
<EVENT id="e2", target="w5"
pred="rain"/>
<TLINK eventID="e1",
relatedToEvent="e2",
relType="BEFORE",
signalID="s1"/>
```

TLINK can be represented in UMR at its document level, as shown earlier in Example 3.

3.3 Quantification and Scope

Pustejovsky et al. (2019) demonstrated how quantifier scoping in ISO-Space could be treated in UMR. Example 5 shows how ISO (2014) annotates quantifier scoping.

(5) a. A computer$_{se1}$ is on$_{ss1}$ every desk$_{se2}$.

b.
```
<spatialEntity
id="se1" pred="computer"
quant="1">
<spatialEntity id="se2"
pred="desk" quant="every"
scopes="se1"/>
<event id="e1"
pred="isLocated"/>
<sRelation id="sr1"
pred="on"/>
<qsLink figure="se1"
ground="se2"
relType="on", trigger="sr1"/>
<scopeLink figure="se2"
ground="se1"
relType="wider"/>
```

The attribute @scopes in <spatial Entity id="se2"/> is not an inherent property of entities but is contextually marked up.

In Example 6, UMR represents quantifier scoping at the document, better called *discourse*, level.

(6) Quantifier Scoping in UMR:

```
(s / sentence
 :text "A Computer is on every desk"
  (i / be-located-at-91
    :theme (c / computer
             :quantity 1)
    :location (d / desk
                :quantity every)))
(d/ discourse level
   :scope (sc :wide sd))
```

The last line in (6), following the UMR guidelines, is to be interpreted as follows: sc indexes the argument of sentence s denoted by c, i.e., *a computer*, while sd indexes the argument of sentence *s* denoted by , i.e., *every desk*. This then can be paraphrased as "every desk (sd) has a wide scope over a computer (sc)".

In the UMR format, Gysel et al. (2021) treats scope by introducing an inverse relation *pred-of* that indicates a predicate like *answer-01* as in Example 7 is a predicate under the scope node.

(7) "Someone didn't answer all the questions."

```
(a /answer-01
   :ARG0 (p /person)
   :ARG1 (q /question
           :quant All :polarity -)
   :pred-of (s / scope
              :ARG0 p :ARG1 q))
```

The scope node indicates that *someone* takes wider scope over *(not) all the questions*.

As in Example 6, we may also represent the scopal relation in Example 7 at the document, better called *discourse* level of UMR.

(8) "Someone didn't answer all the questions."

```
(a /answer-01
   :ARG0 (p /person)
   :ARG1 (q /question
           :quant All :polarity -))
(d /discourse level
   :scope (q :wide p))
```

Unlike Representation 7, Representation 8 explicitly states *someone* p has wide scope over *(not) all the questions* q. Such a discourse-level representation can thus accommodate other types of scopal relations, *dual* and *equal*, which Bunt et al. (2018) claim to be necessary for quantification in general.

With the scopal relations thus specified, Representations 7 and 8 both yield an identical first-order logical form, yielding an identical interpretation:

(9) $\exists p[person(p) \wedge \neg\forall q[question(q) \rightarrow \exists a[answer\text{-}01(a) \wedge \text{ARG0}(a,p) \wedge \text{ARG1}(a,q)]]]$

3.4 Treating Non-consuming Tags

SpatialML (MITRE, 2010), from which ISO-Space originated, introduces so-called *non-consuming tags* for assumed places.

(10) a. Raw Data:
We drove 50 miles east of Boston. The next day, we drove 100 miles north.[1]

b. Three Non-consuming PLACE Tags:
We drove $\text{PLACE}_{pl1:target}$ 50 miles east of $\text{Boston}_{pl2:source}$. The next day, we drove $\text{PLACE}_{pl1:source}$ $\text{PLACE}_{pl3:target}$ 100 miles north.

[1]Taken from MITRE (2010), Section 15.

c. RLINK in SpatialML:

```
<RLINK id=5 source=pl2:Boston
target=pl1 distance=2:50 miles
direction=E signals=2 3/>
<RLINK id=9 source=pl1 target=pl3
distance=6:100 miles
direction=N signals=6 7
```

We can identify a non-consuming tag as an implicit argument to a relation (e.g., an event) that is not syntactically realized.

Every motion triggers a trajectory that a moving object traverses. ISO-Space (ISO, 2020) has thus introduced a non-consuming tag, called *event path*, for trajectories to replace RLINK in SpatialML (MITRE, 2010). Consider Example 11 to see how it is annotated by ISO (2020).

(11) a. Categorized word-segmented Data:
$\text{John}_{x1:w1}$ $\text{drove}_{e1:w2}$ 50_{w3} miles_{w4} east_{w5} of_{w6} $\text{Boston}_{pl1:w7}$.
$\emptyset_{pl2:goal}$ $\emptyset_{ep1}$

b. entity structures:

```
<ENTITY id="x1" target="w1"
type="PERSON" name="John"/>
<EVENT id="e1" target="w2"
pred="DRIVE"/>
<PLACE id="pl1" target="w7"
type="CITY" name="Boston"/>
<PLACE id="pl2"/>
<EVENT_PATH id=ep1 mover="x1"
source="pl1" goal="pl2"
direction="E" distance="50 mi"
trigger="m1"/>
```

c. Link structure:

```
<MOVELINK figure="x1"
ground="ep1"
reltype="TRAVERSES"/>
```

Annotation 11 contains two non-consuming tags: $\emptyset_{pl2:goal}$ and $\emptyset_{ep1}$. The first tag refers to the goal, the second one to the event path created by the motion of John's driving.

Example 12 shows how these non-consuming tags are represented in UMR.

(12) Representing an event-path in UMR:

```
Data (John drove 50 miles east of Boston.)
Predciate-structure level
 (d / drive-01
   :ARG0 (p / person
```

```
          :name (n / name
                :op1 "John"))
   :distance (q / distance-quantity
              :quant 50
              :unit (m / mile))
   :direction (e / east )
   :source (c / city
            :name (n2 / name
                  :op1 "Boston"))
   :goal (p1 / place)
   :path (p2 / path
          :dynamic
          :trigger d
          :mover p
          :source c
          :goal p1
          :distance q
          :direction e)
    :aspect Performance
    :modstr FllAff))
Discourse-structure level
   :moveLink (t1 /traverse
                :arg1 p
                :arg2 p2
                :trigger d))
```

As shown at the discourse-structure level of Example 12 above, UMR successfully represents the traversal relation between the mover p John and the event-path p2 triggered by the motion d of John's driving.

3.5 Complex entity structures

In ISO (2025), some entity structures are annotated as referring to other entity structures to specify their temporal values. Here is an example:

(13) a. Data:
We left$_{e1}$ [$_{t11}$ two weeks]$_{t12}$ before Christmas$_{t2}$.

b. Annotation scheme=ISO (2012):

```
<EVENT id="e1" pred="LEAVE"/>
<TIMEX3 id="t1"
target="two weeks"
type="DURATION" value="P2W"
beginPoint="t11" endPoint="t2"/>
<TIMEX3 id="t12"
type="DATE" value="2004-12-25"/>
<TIMEX3 id="t11"
type="DATE" value="2004-12-11"
temporalFunction="TRUE"
anchorTimeID="t1"/>
<EVENT id="e2" pred="Christmas"/>
<TLINK eventID="e1"
relatedToTime="t11"
relType="IS_INCLUDED"/>
<TLINK eventID="e2"
relatedToTime="t12"
relType="IDENTITY"/>
```

The entity structure <TIMEX3 id=t1> in (13b) has two attributes, @beginPoint and @endPoint, which refer to other entity structures for their values. The value of @beginPoint is calculated as 2024-12-11, anchored to the Christmas day t1, as annotated in <TIMEX3 id=t11> with two attributes @temporalFunction and anchorTimeID.

AMR can also represent how the value of @beginPoint of a time interval, on which the motion of "our levaing" took place, is expressed:

(14)

```
Data (We left two weeks before Christmas.)
Predicate-structure level
 (l/ left-01
   :ARG0 (p / person
             :ref-person 1st
             :ref-number Plural)
   :time (d / date-entity
            :mod (t3 / temporal-interval
               :quant 2 :unit (w / week))
                :start (d1 / date-entity
                              :month 12
                              :day 11)
                :end (d2 / date-entity))
   :temporal (b/ before
               :op1 (n/ name
               :op2 (c/ Christmas
                  :date (d2 / date-entity
                              :month 12
                              :day 25))))
   :aspect Performance
   :modstr FllAff)
Discourse-structure level
   :corefence (s / same-date
                 :arg1 d
                 :arg2 d1)
   :temporal (c / contains
                 :arg1 d
                 :arg2 l))
```

On the entity structure level, the start of the 2-week duration is dated December 11, for the end of the duration is the same date of Christmas, December 25, as represented on the link structure level.

The departure is also represented as occurring on December 11 at the link structure level.

4 Motion-oriented Dynamic Information

4.1 Overview

Pustejovsky and Moszkowicz (2011) combined TimeML (Pustejovsky et al., 2005) and SpatialML (Mani et al., 2010) into the Spatio-temporal Markup Language (STML) to annotate dynamic information involving motions and motion paths in language. Now, STML can be updated to ISO (2012) and ISO (2020), which have formally defined the notion of event paths triggered by motions. An event path, triggered by a motion, is traversed by a moving object and is thus defined as a nonempty finite directed sequence of spatio-temporally delimited positions of a moving object. Dynamic Interval Temporal Logic (DITL) was adopted as the semantics of STML for reasoning with programs.

We work with an excerpt from a travelogue through Central America, taken from Pustejovsky and Moszkowicz (2011):

(15) Sample Raw Data:
John left San Cristobal de Las Casas four days ago. He arrived in Ocosingo that day. The next day, John biked to Agua Azul and played in the waterfalls for 4 hours. He spent the next day at the ruins of Palenque and drove to the border with Guatemala the following day.

We first show, in Subsection 4.2, how STML annotations in XML are represented in UMR.

4.2 Representing STML Annotations in UMR

For illustration, we take the first sentence from Data 15 and segment it into words and mark up their category identifiers.

(16) Sample Data: Categorized Segmentation
$_{S1}$[John$_{se1:w1}$ left$_{m1:w2}$ [San Cristobal de Las Casas]$_{pl1:w3}$ four days$_{t1:w4-5}$ ago$_{s1:w6}$].

We now apply STML to annotate Sample Data 16 in XML.

(17)
```
<annotation id="a1" aScheme="STML">
  <spatialEntity id="se1" target="w1"
    type="person" name="John"/>
  <motion id="m1" target="w2"
    type="transition" pred="leave"/>
  <place id="pl1" target="w3"
    cvt="town" form="name"/>
  <timeX3 id="t1" target="w4-5"
    type="duration"
    value="4" unit="day"
    start="t11" end="t12"/>
  <timeX3 id=""t11"target=""
    type="date" value="2025-03-08"/>
  <timeX3 id="t12" target=""
    type="date" value="2025-03-12"
    trigger="s1"/>
  <signal id="s1" target="w6:ago"/>
  <eventPath id="ep1" target=""
    start="<pl1,t11>"
    end="<unknown,t12>" trigger="m1"/>
  <tLink id="tL1" eventID="m1"
    relatedToTime="t11"
    relType="DURING"/>
  <moveLink id="mvL1" figure="se1"
  ground="ep1" relType="traverses"/>
  </annotation>
```

Annotation 17 above represents the information about John's departure from San Cristobal, which occurred on the day marked as t11. This date represents part of the mover's start position <pl1, t11> of a 4-day duration or interval stretched to the present utterance time, today or DCT (document creation time).

Representation 18 now shows how Annotation 17 in XML can convert to UMR:

(18) Data (John left San Cristobal de Las Casas four days ago.)

```
Predicate-structure Level
(l / leave-01
  :ARG0 (s1p / person :name John)
  :time (d / date-entity
          :mod (t1 / temporal-interval
                  :duration (v / value
                               :quant 4
                               :unit day)
                  :start (d1 / date-entity
                               :year 2025
                               :month 3
                               :day 8)
                  :end (t2 / today)))
  :source (s / start-position
            :op1 (l2 / location
                    :name San Cristobal
                          de Las Casas)
```

```
            :op2 d1)
  :aspect Incremental Accomplishment
  :modstr FllAff)
Discourse-structure Level
 (:temporal (b / before
               :arg1 d :arg2 t2)
  :temporal (c / contains
               :arg1 d :arg2 l))
```

John's departure implies a durative performance of eventually reaching a goal. This action also develops incrementally. Hence, UMR marked the aspect of `leave-01` as *Incremental Accomplishment* in UMR Representation 18, while the ISO annotation schemes fail to do so.

Temporal Interval vs Duration In Example 18, the concept `:time` refers to the occurrence time of the motion *leave*, whereas the concept `:duration` is its modifier. In Example 19, on the other hand, the duration *four hours* modifies John's activity of playing directly, meaning that it lasted four hours, while *the next day* was the time of its occurrence.

(19) Data: (The next day, John biked to Agua Azul and played in the waterfalls for 4 hours.)

```
Predicate-structure Level
(b \ bike
   :ARG0 John
   :time (d / day)
   :duration (t / temporal-quantity
                :quantity 4 unit:day))
```

4.3 Adopting GLAMR

Tu et al. (2024) propose a Generative Lexicon-based AMR (GLAMR) to capture the dynamics associated with change predicates. Adopting GL's subevent structure for verb meaning (Pustejovsky, 1995), a predicate meaning consists of a series of subevent structures related to various transitions triggered by motions or transactions, such as transfer of possessions as in GL-VerbNet (Brown et al., 2019). This structure provides relevant spatio-temporal information on sub-event structures related to various transitions. It also captures the aspectual notions of incremental accomplishment by adding the event structure directly under the topic predicate node, as in Example 20.

(20) `t / target (John left San Cristobal de Las Casas four days ago.)`

```
Predicate-structure level
  (l/ leave-01
     :ARG0 (j / john)
     :event-structure (s /subevents
       :E0 (d / do
             :action l)
       :E1 (h / has_position
              :theme j
        :initial_loc (s1 / San Cristobal)
           :initial_time d1)
       :E2 (a / and
           :op1 (m / motion
              :moving-object j
              :trajectory p)
           :op2 (h1 / has_position)
              :polarity -
              :theme j
              :location s2
              :time d2))
     :time (d / date-entity
            :mod (t1 / temporal-interval
           :duration (q1 / temp-quantity
                         :quantity 4
                       :unit (d3 / day))
              :start (d1 / date-entity)
              :end (t3 / today)))
     :event-path (p / positions
        :trigger m
        :moving-object j
        :start (p1 / position
          :location s1
          :time d1
            :op1 (q2 / spatial-quantity
                  :unit meter
                  :quantity 0))
        :next (p2 / position
          :location s2
          :time d1
           :op1 (q3 / spatial-quantity))
        :end (p3 / position))
     :modstr FllAff)

Discourse-structure level
     :temporal (b / before
                 :arg1 d1 :arg2 t3)
     :spatial  (g / greaterThan
                 :arg1 q3 :arg2 q2))
```

The event-structure and the event-path share values, but from different perspectives. The sub-event `E2` triggers the `event-path` as a trajectory of a moving object j. John's position changed as he moved: he was no longer in San Cristobal's initial location `s1` but moved to the next location `s2`, while all these

sub-events occurred on the same day.

At the discourse or link structure level, two relations are represented: temporal and spatial. The temporal relation states that the day d1 of John's departure from San Cristobal preceded the DCT t3, today, while the duration says there was a four-day interval between the departure day and the DCT. The spatial relation then states that the event-path length has lengthened from q2 to q3 while the mover moved from the start location s1 to the next location s2 or s1+1.

4.4 Applying Dynamic Interval Temporal Logic

DITL[2] formalizes the dynamic aspectual notion of incremental accomplishment in UMR as a program in DITL. Pustejovsky and Moszkowicz (2011) (page 16) formulates the notion of a directed motion leaving a trail as a program, represented with minor modifications in DITL, as in:

(21) Motion Leaving a Trail:
$move_{tr}(x) =_{df} pos(x) := y, b := y,$
$p := (b); (y := z, y \neq z, p := (p, z))^{+}$

This program states that the trail path p stretches as the beginning point b of the mover x by the Kleene iteration $+$ (more than one occurrence), as the mover x moves on. Then, the motion-triggered dynamic path p will be a sequence of x's positions, incremented iteratively as time progresses. Here, the notion of position $pos(x)$, defined as a complex function from time to $loc(x)$, which is the location of a moving object x, replaces the notion of $loc(x)$.

4.5 Dynamic Space as Minimal Embedding Ground

The spaces in which dynamic paths stretch out are also constrained by their embedding ground. Climbing over a hill creates a path tangential to the surface shape of the hill. In contrast, flying over a hill may create a path almost tangential but detached from it.

(22) Minimal Embedding Grounds
a. John climbed *over* the hill.
b. The helicopter flew *over* the hill.
c. Joh swam *around* the lake.
d. John walked *around* the lake.

Swimming around a lake means it takes in the water, whereas running around the lake means a circular activity outside the lake. Despite the same use of spatial relators like *over* and *around*, each action or activity is characterized by a different embedding ground. Hence, the fine-grained characterization of motions or their paths should be specified with the type of embedding ground in both ISO semantic annotations and UMR.

5 Concluding Remarks

There are two commonalities between ISO SemAF standards and UMR. First, both ISO-TimeML and ISO-Space emphasize the role of events and motions. Such a focus fits well into the structure of AMR and UMR, both of which stress the predicative core of propositional content.

Second, the dual annotation structure of ISO semantic annotation frameworks such as ISO-TimeML and ISO-Space conforms perfectly to the dual level of UMR, sentence (predicate structure)-level and document (discourse)-level.

There are, however, some differences. First, ISO SemAF uses a semantic role link, tagged SRLINK, to assign participant semantic roles to events. By following neo-Davidsonian semantics, AMR/UMR treats them as relations between event instances and their arguments or adjuncts. ISO's semantic link needs to be applied repeatedly to assign a series of participant roles. AMR/UMR, in contrast, directly copies a series of those roles associated with each predicate from available linguistic resources such as PropBank.

Secondly, the degree of granularity in AMR /UMR differs from ISO SemAF in treating dialogue acts, discourses, and quantification. Such differences can, however, be fixed with minor but time-consuming modifications. AMR/UMR requires additional structural modifications to represent dialogue and discourse structure in a richer and more expressive fashion, one accommodating the needs of dialogue and discourse understanding in NLP. Developing such further extensions to UMR based on the work carried out within the ISO working group is an exciting challenge, and promises to better integrate standards specifications within the family of AMR representations.

[2] Mani and Pustejovsky (2012) has a fuller version of introducing DITL.

Limitations

The scope of this paper is restricted. It mainly compares the representation of two ISO SemAF standards, ISO-TimeML and ISO-Space, with UMR. Our future work should be extended to other ISO standards on dialogues, discourses, quantification, and quantitative information in general. It should include studying details in annotating the tense, aspect, and modality of predicates, the specification of which varies much from language to language.

We have intentionally avoided evaluating UMR. We have accepted the review by Bos (2016) for its semantic adequacy and some articles, such as Van Gysel et al. (2021), for learnability, scalability, or applicability to computing applications. This paper did not compare computational application or scalability between ISO SemAF and AMR/UMR. This is mainly because ISO SemAF has focused on the abstract and theoretical formulation of semantic annotation structures rather than on issues of direct use in industrial applications.

We have not yet experimented with the possibility of amalgamating UMR with DRT or its subsequent extensions for semantic representation. One interesting proposal is to treat events like *walk* not as a functional type $e \rightarrow t$ but a basic type e in DRSs. We then have $[instance(e, walk), instance(j, John), actor(e, j)]$ in DRS as well as in UMR, instead of $[walk(e), John(x), actor(e, x)]$ in DRT. With this proposal accepted, we think the UMR logical format and the DRT representation format are identical.

The focus of this paper on attempting to convert XML-represented annotations to AMR/UMR is motivated by the fact that most of the ISO SemAF standards use XML as their representation format (although the DialogueBank (Bunt et al., 2016), a multilingual resource of dialogues annotated according to ISO 24617-2:2012 also uses two alternative representation formats and supports the conversion among them.) This has made all ISO SemAF standards interoperable with other ISO annotation standards on the other linguistic levels, such as lexicology, morphology, syntax, and data construction, all based on XML and the TEI Guidelines for using XML for text processing.

We understand UMR is at a developing stage and may remain as such. Our ISO working group on semantic annotation believes that some of our standards cover semantic issues such as dialogues, discourse theories, and quantification in much more breadth and depth and hopes to contribute to the editing of UMR guidelines in the future. The ISO semantics group will learn much in the area of computational applications through continued interactions with the UMR group.

Ethics Statement

All authors believe this work contributes to advancing natural language understanding, enabling more accurate and robust analysis of human-produced text. We collectively hope it will help expand equitable access to information and improve Human-Computer Interaction. At the same time, we emphasize the need for ongoing monitoring of societal impacts, particularly regarding the potential amplification of harmful stereotypes or disinformation.

Acknowledgements

The authors would like to express their gratitude to everyone whose efforts made this work possible. In particular, the lead author, Kiyong Lee, extends sincere appreciation to his co-authors for their insightful discussions and countless hours spent refining the manuscript. We are likewise deeply indebted to the three anonymous reviewers and the decision committee, whose careful reading and thoughtful critiques greatly enhanced the clarity and overall quality of the paper. Any remaining shortcomings are, of course, our own.

References

Johan Bos. 2016. Squib: Expressive power of abstract meaning representations. *Computational Linguistics*, 42(3):527–535.

Susan Windisch Brown, Julia Bonn, James Gung, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2019. VerbNet representations: Subevent semantics for transfer verbs. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 154–163, Florence, Italy. Association for Computational Linguistics.

Harry Bunt. 2010. A methodology for designing semantic annotation languages exploiting syntactic-semantic iso-morphisms. In *Proceedings of ICGL 2010, the Second International Conference on Global Interoperability for Language Resources*, pages 29–45, City University of Hong Kong.

Harry Bunt, Volha Petukhova, Andrei Malchanau, Kars Wijnhoven, and Alex Fang. 2016. The DialogBank. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*,

pages 3151–3158, Portorož, Slovenia. European Language Resources Association (ELRA).

Harry Bunt, James Pustejovsky, and Kiyong Lee. 2018. Towards an iso standard for the annotation of quantification. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, pages 1787–17949, Miyazaki, Japan. ELRA.

Van Gysel, E. L. Jens, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andres Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James H. Martin, Stephan Oepen, Martha Palmer, James Pustjovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35:343–360.

ISO. 2012. *ISO 24617-1:2012, Language resource management – Semantic annotation framework (SemAF) – Part 1: Time and events*. The International Organization for Standardization, Geneva. Editorial committee: James Pustejovsky (chair), Branimir Boguraev, Harry Bunt, Nancy Ide, and Kiyong Lee (project leader).

ISO. 2014. *ISO 24617-7:2014 Language resource management – Semantic annotation framework (SemAF) – Part 7: Spatial information*, 1st edition. The International Organization for Standardization, Geneva. Project leaders: James Pustejovsky and Kiyong Lee.

ISO. 2020. *ISO 24617-7:2020 Language resource management – Semantic annotation framework (SemAF) – Part 7: Spatial information*, 2nd edition. The International Organization for Standardization, Geneva. Project leaders: James Pustejovsky and Kiyong Lee.

ISO. 2025. *ISO 24617-12:2025 Language resource management – Semantic annotation framework (SemAF) – Part 12: Quantification*. The International Organization for Standardization, Geneva. Project leader: Harry Bunt.

Kiyong Lee. 2023. *Annotation-Based Semantics for Space and Time in Language*. Cambridge University Press, Cambridge, UK.

Inderjeet Mani, Christy Doran, Dave Harris, Janet Hitzeman, Rob Quimby, Justin Richer, Ben Wellner, Scott Mardis, and Seamus Clancy. 2010. SpatialML: annotation scheme, resources, and evaluation. *Language Resources and Evaluation*, 44:263–280.

Inderjeet Mani and James Pustejovsky. 2012. *Interpreting Motion: Grounded Representation for Spatial Language*. Oxford University Press, Oxford, UK.

MITRE. 2010. *SpatialML 3.1: Annotation Scheme for Marking Spatial Expressions in Natural Language*. The MITRE Corporation, cdoran@Dmitre.org.

James Pustejovsky. 1995. *Generative Lexicon*. The MIT Press, Cambridge, MA.

James Pustejovsky. 2017a. ISO-TimeML and the annotation of temporal information. In *Handbook of Linguistic Annotation*, volume 2, pages 941–988.

James Pustejovsky. 2017b. ISO-Space: Annotating static and dynamic spatial information. In *Handbook of Linguistic Annotation*, volume 2, pages 989–1024.

James Pustejovsky, Robert Ingria, Roser Saurí, Joseé Casta no, Jessica Littman, Rob Gaizauska, Andreas Setzer, Graham Katz, and Inderjeet Mani. 2005. The specification language TimeML. In *The Language of Time: A Reader*, pages 545–557, Oxford. Oxford University Press.

James Pustejovsky and Jessica L. Moszkowicz. 2011. SpatialML: annotation scheme, resources, and evaluation. *Language Resources and Evaluation*, 44:263–280.

James Pustejovsky, Nianwen Xue, and Kenneth Lai. 2019. Modeling quantification and scope in abstract meaning representations. In *Proceedings of the First International Workshop on Designing Meaning Representation*, pages 28–33. Association for Computational Linguistics. Florence, Italy, August 1, 2019.

Jingxuan Tu, Timothy Obiso, Bingyang Ye, Kyeongmin Rim, Keer Xu, Liulu Yue, Susan Windisch Brown, Martha Palmer, and James Pustejovsky. 2024. GLAMR: Augmenting AMR with GL-VerbNet event structure. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7746–7759, Torino, Itay. ELRA and ICCL.

Working Group UMR. 2022. *Uniform Meaning Representation (UMR) 0.9 Specification*. UMR Working Group for Guidelines.

Jens E. L. Van Gysel, Meagan Vigus, Lukas Denk, Andrew Cowell, Rosa Vallejos, Tim O'Gorman, and William Croft. 2021. Theoretical and practical issues in the semantic annotation of four indigenous languages. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 12–22, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Author Index